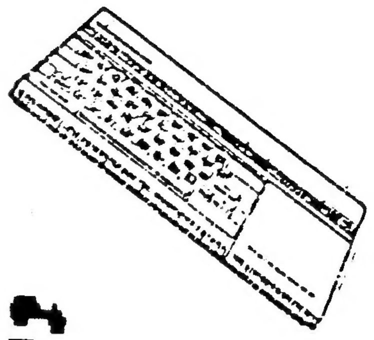
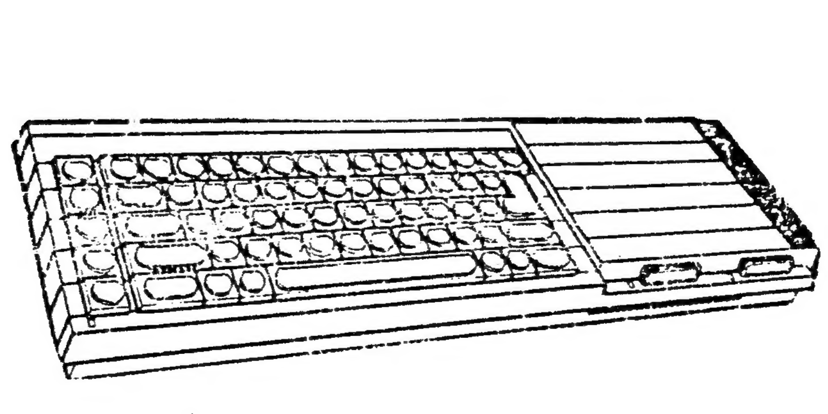
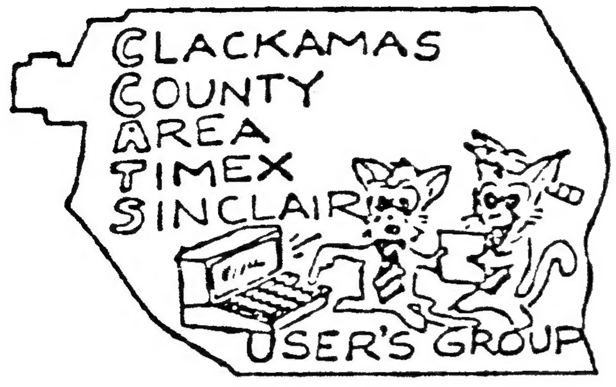
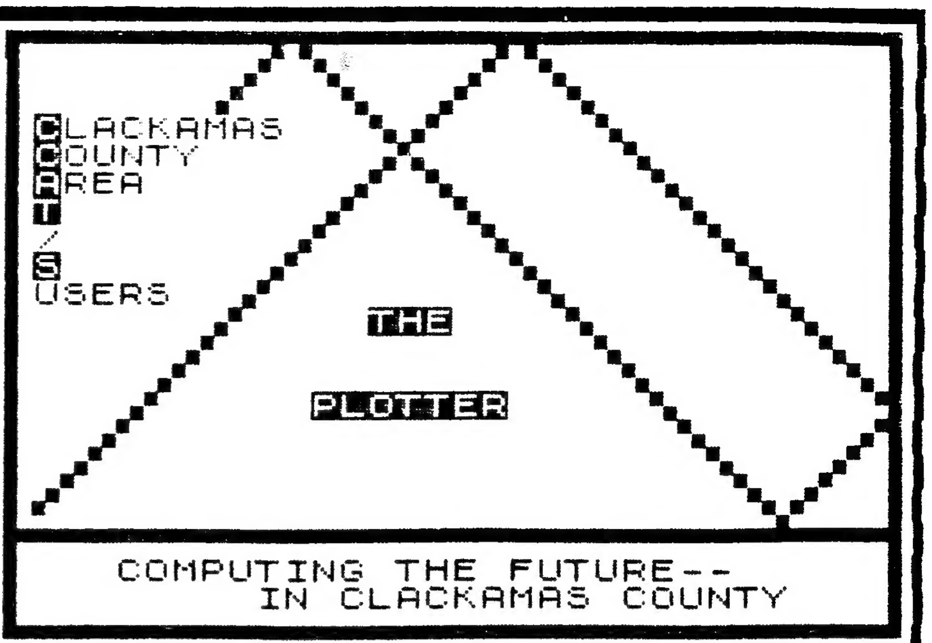


CCATS PRESENTS

THE BEST OF THE PLOTTER

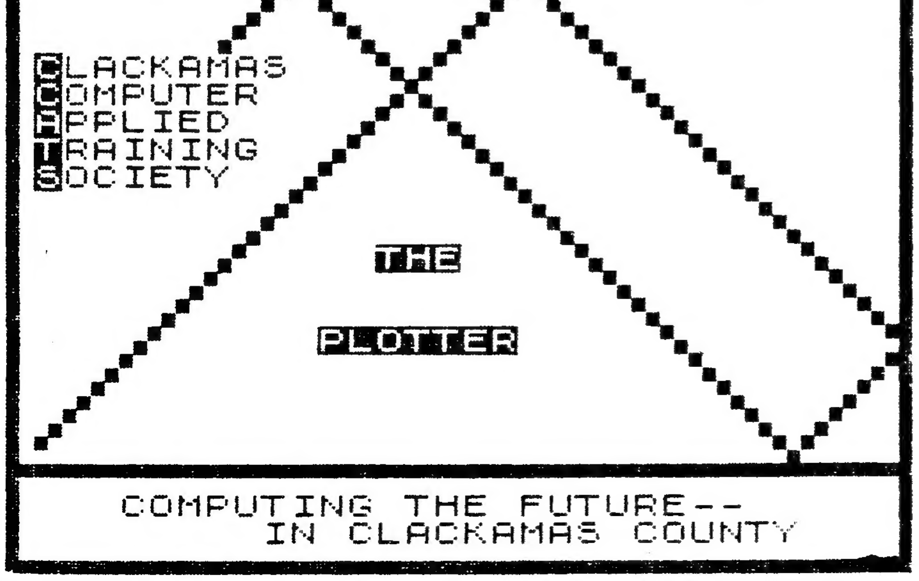


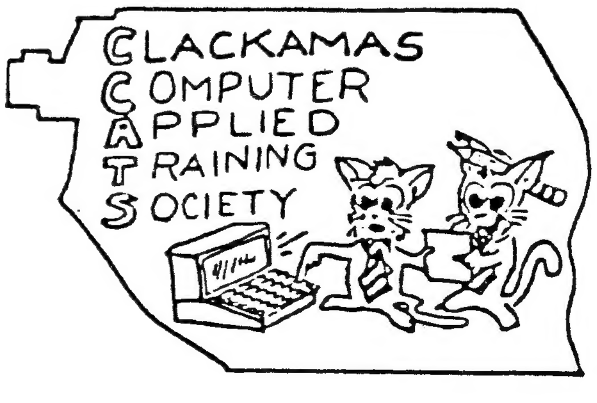




From Our Files-
We Proudly Present:

The Best OF
THE PLOTTER





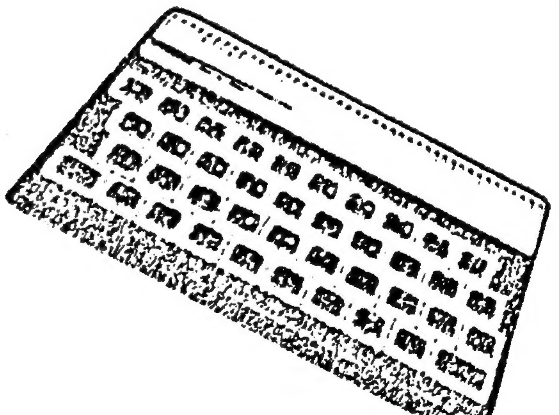
Published by:
RMG Enterprises
14784 S. Quail Grove Circle
Oregon City, OR 97045

The Best Of The Plotter

Table Of Contents:

Section 1:

ZX-81 T/S1000 T/S1500



16K RAMpack Test program

Type in and RUN...

```
   1 POKE 18000,33
   2 POKE 18001,11
   3 POKE 18002,0
   4 POKE 18003,57
   5 POKE 18004,68
   6 POKE 18005,77
   7 POKE 18006,201
   8 PRINT USR (18000-16373)/102
4:"K"
```

PROGRAMS AND ROUTINES...

Here is the first of our series on JUSTIFICATION. It was submitted by Dennis Jurries, who says that he would like to have a short session at the next meeting for those who are interested, to go into further explanation on this subject. Be sure to attend.

RIGHT JUSTIFICATION...

```
  10 LET A$="YOUR NUMBER"
  20 PRINT AT X,Y-LEN A$;A$
```

INPUT WITHOUT CURSOR...

Have you ever wanted to put together a routine in which the computer would wait for an input WITHOUT a CURSOR? If so, try this:

```
  1000 IF INKEY$<>"5" THEN GOTO 10
  00
  1010 PRINT AT 10,10; "5"
```

The INKEY$ function does not actually wait for a key to be pressed, rather it scans the keyboard for a certain key being pressed, then takes appropriate action. More on this can be found in the manual under TIME and MOTION.

1K ADVENTURE

by David Clark

NOTE: As you can see, this program makes good use of the INKEY$ function. That is the reason I decided to run it at this time. In future issues there will be other programs reprinted from magazines that are not in the club library.

```
   5  REM  1K ADVENTURE
  10 LET S=VAL "50"
  15 LET T=PI-PI
  20 LET TU=PI/PI
  25 PRINT "TURN ";TU;" TREASURE
";T;" STRENGTH ";S
  30 IF S<=0 THEN  GO TO 5000
  35 PRINT "YOU HAVE FOUND:"
  40 LET X=INT (RND*VAL "75")
  45 LET H=INT (RND*INT PI)+PI/PI

  65 GO SUB H*100
  70 LET S=S+10
  80 CLS
  90 LET TU=TU+1
  95 GO TO VAL "25"
 100 PRINT "A MONSTER. F/B?"
 110 INPUT I$
 120 IF I$<>"F" AND I$<>"B" OR I$
="B" AND T<X THEN  GO TO 110
 130 IF I$="F" THEN  LET S=S-X
 140 IF I$="B" THEN  LET T=T-X
 150 GO SUB 2000
 190 RETURN
 200 PRINT "A POTION"
 210 LET S=S+X
 220 GO SUB VAL "2000"
 230 RETURN
 300 PRINT "TREASURE. IT IS WORTH
 ";X
 310 LET T=T+X
 350 GO SUB VAL "2000"
 360 RETURN
2000 IF INKEY$="" THEN  GO TO 200
0
2010 RETURN
5000 CLS
5010 PRINT "YOU HAVE DIED.",,"YOU
 SURVIVED ";TU;" TURNS"
5100 STOP
5110 SAVE "1KADVEN"
5120 RUN
5130 STOP
5150 RANDOMIZE USR 100: SAVE "1ka
dvn.B1" LINE 1
```

HOW TO CHANGE RAMTOP...PART ONE

by Dick Wagner

Part one is the blind faith part of this article, part two will be a discussion on the background used for part one.

Certain programs call for RAMTOP to be lowered, sometimes by a certain number of bytes. I recently ran into this problem, so I decided that it was something that I had to know more about.

With 16K of memory, our computers have a RAMTOP address of 32767. Computers with 2K of memory have an address of 18432 for RAMTOP. The 1K RAM in the ZX81 has a RAMTOP address of 17408.

The following direct command will ask the computer for the RAMTOP address in your computer:

```
PRINT PEEK 16388+256*PEEK 16389
```

The easiest method to lower RAMTOP is based on using increments of 256 to determine the amount of memory to leave above RAMTOP. Consider that 4*256=1024=1K and 256=1/4K. With this method two numbers must be POKE'd, as in:

```
POKE 16388,0 (0=increments of 256)
POKE 16389,N (N=128 for an address
of 32767
```

Where N is number based on this table:

K	N	DIFF. FROM 128
1/4	127	1
1/2	126	2
1	124	4
2	120	8
4	112	16
8	96	32
16	64	64

Select the number N determined by this table of values, where K is the amount to lower RAMTOP. I suggest that you copy this table on a card along with the following example on a card for future reference, or put a copy in your computer notebook.

EXAMPLE:

Lower RAMTOP 2500 or 2 1/2K

```
  2K = 8
1/2K= 2
-------
     10

   N=128-10=118
or N=120(2k)-2(1/2K)=118

Now: POKE 16388,0
     ENTER
     POKE 16389,118
     ENTER
     NEW
     ENTER
```

Blind faith now says that RAMTOP =32768-2 1/2*1024=30208. If you question faith, then check your work with:

```
PRINT PEEK 16388+256*PEEK 16389
```

NOTE: While POKE will accept either the 16388,0 first, or the 16389,118 first. The PEEK's MUST be in the order shown.

CHANGE RAMTOP...PART TWO

The general procedure for lowering RAMTOP in increments of 1/4K (256) was given in Part 1. Here is the reasoning for those who like more than blind faith.

The Sinclair memory system includes a variable upper limit called RAMTOP. When the computer first powers up, the microprocessor, MP, inputs a binary number 2 into each memory cell, or address, in RAM. Then the MP interrogates each address. When it finds an address without a binary 2, MP calls this the top limit of memory. This information is stored in dedicated memory addresses 16388 and 16389. This information can only be changed by actually changing the address of RAMTOP.

Part 1 gave this direct command to determine the present RAMTOP address:

```
PRINT PEEK 16389*256+PEEK 16388
```

This command will cause the address to be printed at 0,0 and will be applicable to your computer memory capacity. If the numbers are smaller than those given in Part 1, either there is a defective memory cell or RAMTOP has already been lowered. To return RAMTOP to standard 32767 address:

```
  POKE 16388,0
  POKE 16389,128
```

Change 128 to 72 for 2K of memory. To confirm the address, PEEK again.

The address 16388 is filled with 8 zeroes in binary. If there are 8, then something is left over for address 16389. If there is only one bit over then it must be 256 as 2 to power of 8 equals 256, and the number to input into 16389 You may notice a RAMTOP address variation such as 32767 or 32768. 32767 is the actual address but to be above by 1, we use 32768. The following formula takes this into account:

```
POKE 16389, INT ((32768-MK)/256)
POKE 16388,(32768-MK)-256*INT((3
2768-MK)/256)
```

...where MK is the amount of memory we wish to store above RAMTOP and is not abbreviated as 2 1/2K (use 2500), etc. This formula will work only for 16K of memory, so substitute if necessary. As mentioned at first, 16389 and 16388 do not vary, only the record in them varies.

The computer is not particular about the two addressed locations being POKE'd in special order as either 16388 or 16389 can be first.

GRAPHIC DEMO ON THE 1000

by Dick Wagner

This month our program also comes from Dick Wagner. This program could be used as an attention getter, as it does have some action. The PLOT pixel looks like it is jumping up and down a flight of steps. Line 10 and 50 PRINT the name and delete the PLOT pixel. This is a simple way to get around using PRINT AT. Try changing lines 20 and 40 to N,0 and lines 70 and 80 to X,0. This shows that PLOT will wipe out characters when using the same space. Then change line 20 and 40 to N,10 to give different action again. Try the original program but change 70 and 80 to X,0. To keep from deleting the bottom name, change line 10 to N=2 TO 43.

```
   1 REM AND EASY METHOD TO MAKE
AN ACTION ATTENTION GETTER.
   2 REM SEE TIMEX USER MANUAL P
AGE 65 ABOUT PLOT MOVES THE PRIN
T POSITION. DID YOU SKIP OVER TH
IS?
   3 REM USE RUN TO START THE AC
TION.
  10 FOR N=1 TO 43
  20 PLOT N,N
  30 PRINT "LAURIE G."
  40 UNPLOT N,N
  50 NEXT N
  60 FOR X=43 TO 1 STEP -1
  70 PLOT X,X
  80 UNPLOT X,X
  90 NEXT X
 100 GOTO 10
 200 REM THE WORD PRINT-OUT STAR
TS AT PLOT 0+3 BECAUSE PLOT CHAR
ACTER IS 1/2 OF PRINT CHARACTER
WIDE. THIS EXERCISE DEMONSTRATES
 REM 2
 300 REM LINES 60-90 PRINTS PLOT
PIXELS IN REVERSE. LINE 100 KEEP
S THE ACTION GOING.
```

```
                              LAURIE G.
                             LAURIE G.
                            LAURIE G.
                           LAURIE G.
                          LAURIE G.
                         LAURIE G.
                        LAURIE G.
                       LAURIE G.
                      LAURIE G.
                     LAURIE G.
                    LAURIE G.
                   LAURIE G.
                  LAURIE G.
                 LAURIE G.
```

CHARACTERS ON THE 1000

by Dick Wagner

PART 1

The 1000 computer does not provide access to the smallest increment or dot (pixel) the computer produces. PLOT characters are made up of 16 dots and use 1/4 of the normal character cell as do several GRAPHIC symbols; these are what we have to work with.

We can however, see how the various characters are made. By using a monitor or TV with good resolution, we can observe the composition of these in the form of dots. The normal space provides for a format of 8x8 dots to make GRAPHIC symbols shown on the keys. However, other characters must provide a surrounding space of at least 1 row on all sides so characters have 2/8 of a space between them. Thus, the letters and numbers are limited to a format of 6x6 dots. to observe this, make up a 3 line program:

```
  10 PRINT " █ "
  20 PRINT "█ █"
  30 PRINT " █ "
```

...where █ is CHR$ 8 GRAPHIC A. A good TV will display CHR$ 8 as 8 rows of dots staggered with 4 dots per row. Thus, 2 rows will display 8 dots.

Change line 20 by putting your favorite character in position 2. Z and M show the clear boundary on all sides. Quotes, periods, commas and other punctuation marks use the fewest dots and our little program will show where they are in relation to the character space. For instance, a period is 2 dots high and 1 dot wide while a "," is 3x2.

The designers of Sinclair BASIC had to work within the limits of the 6x6 format. We can study how the designers elected to make diagonal and curved lines, which is really a challenge to make intelligent marks on the screen.

PART 2

If you tried the example in Part 1, you probably had difficulty in making out each dot. By enlarging the character, we can then see them. The following program from SYNCHRO-SETTE magazine will print most characters 8 times larger. Because of the program structure, not all keyboard characters will work. These enlarged characters will look crude because of being formed with a square character instead of a rounded pixel.

```
  10 LET S=0
  20 PRINT AT 0,0;"ENTER ANY CHA
RACTER..."
  30 INPUT A$
  40 LET P=7
  50 LET F=CODE A$
  60 LET F=8*F-8
 100 FOR B=7688 TO 7688+F+7
 110 LET A=PEEK B
 120 FOR I=7 TO 0 STEP-1
 130 PRINT AT 21-P, I+S; CHR$ ((
A-2*INT (A/2))*128)
 140 LET A=INT (A/2)
 150 NEXT I
 160 LET P=P+1
 170 NEXT B
 180 LET S=S+8
 190 IF S>25 THEN GOTO 300
 200 GOTO 20
 300 FOR N=1 TO 8
 310 SCROLL
 320 NEXT N
 330 GOTO 10
9998 SAVE "BIG/CHAR EXAM"
9999 RUN
```

A MACHINE LOADER PROGRAM FOR T/S 1000

by Dick Wagner

This machine language loading program for loading REM lines used Gordon Young's article "Build Your Own Spreadsheet" in SYNC Jan/Feb 1984 issue. The method provides for entering the MC, one at a time, into a table 6 columns across, left to right. The MC number can be edited when first displayed, and then again in the table (last line only and not the 6th column). The MC listing scrolls so you have unlimited display space.

It is possible to SAVE the REM listing while in process! The LOADed MC listing (incomplete) can be continued until finished. If STOP is used before finishing, any further input starts back at the beginning. Accordingly you cannot see characters going into REM until finished.

If you have a different column count than 6, then change 9935, 9940 & 9975. The M<26 assures the loops continue and when M>26, the loop stops on the 5th input. The M+5 and M-5 allows for 5 inputs of 3 characters and 2 spaces each plus the last 3 characters. See Gordon Young's article for complete details. 16514 is the address for the first byte after REM. This program is for 60 rows of data in 6 columns.

```
9900 LET N=16514
9905 LET M=0
9910 INPUT A
9915 IF A>300 THEN GOTO 9960
9920 POKE N,A
9925 PRINT AT 20,M;A
9930 LET N=N+1
9935 LET M=M+5
9940 IF M<26 THEN GOTO 9910
9945 LET M=0
9950 SCROLL
9955 GOTO 9910
9960 IF A>555 THEN GOTO 9999
9965 IF A=5 THEN GOTO 9990
9970 LET N=N+1
9975 IF M<26 THEN LET M=M-5
9980 PRINT AT 20,M;"    "
9985 GOTO 9910
9990 SAVE "ASSEMBLER"
9995 GOTO 9910
9999 STOP
```

SAVE CHR$ USR 832"PROGRAM NAME"

This one line routine will help you make a copy of those unsaveable programs. Put the tape you want to copy in your recorder, make sure that you have a new, blank tape in front of your recorder, as you have only five, that's 5, seconds in which to turn off the recorder, remove one tape, insert the new tape, and start the recorder on "record". That is not a lot of time as you will find out, but, with a few practice runs you should have no trouble. This routine tells the computer to load the program and then, via a USR call, tells the machine to save it exactly as it was loaded, and, if there are no bugs in the program, the routine will work every time. Give it a try.

MATH DRILLS

by Rod Gowen

This month for our program section we thought be would try to show some of you newer programmers what can be accomplished by a novice hacker with only 1K of RAM to work with.

These three simple math routines were the very first programs written by yours truly right after reading the ZX81 manual. The routines were designed to help my daughters in routine MATH SKILLS. They really did help. They also show that some nice things can be accomplished with very little knowhow and/or memory.

```
 10>REM MATH DRILL M
 15 REM Multipication drill
        Written by-
      Rod Gowen
 20 REM SLOW goes here for 1000
```

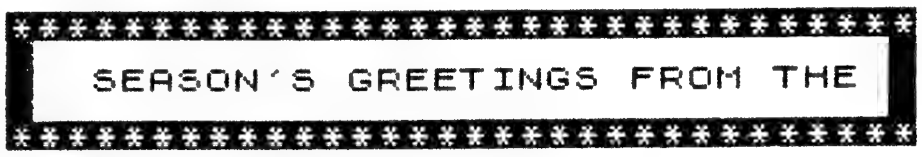
```
  30 CLS : PRINT "MULTIPLICATION
DRILL"
  40 PRINT
  50 LET X=INT (RND*20)+1
  60 LET Y=INT (RND*20)+1
  70 PRINT AT 10,0;X;"  TIMES  ";
Y;"=";,"ANSWER?"
  80 INPUT Z
  90 PRINT AT 10,15;,Z;"      "
 100 PAUSE 200
 110 CLS
 120 IF Z=X*Y THEN  PRINT AT 5,0;
"GREAT  *****  SUPER";,,,;"NEXT?"

 130 IF Z=X*Y THEN  GO TO 50
 140 IF Z<>X*Y THEN  PRINT AT 5,0
;"SORRY, WRONG ANSWER-",,,,"TRY A
GAIN"
 150 PRINT
 160 IF Z<>X*Y THEN  GO TO 70
 170 CLS
 180 STOP
 200 CLS : REM MATH DRILL A
 210 REM Addition Drill
        Written by-
       Rod Gowen
 220 REM SLOW goes here for 1000
 230 PRINT "ADDITION DRILL"
 240 PRINT
 250 LET X=INT (RND*500)+1
 260 LET Y=INT (RND*500)+1
 270 PRINT AT 10,0;X;"  PLUS  ";Y
;"=";,"ANSWER?"
 280 INPUT Z
 290 PRINT AT 10,15;,Z;"      "
 300 PAUSE 200
 310 CLS
 320 IF Z=X+Y THEN  PRINT AT 5,0;
"HOORAY!  =====  GO ON-"
 330 IF Z=X+Y THEN  GO TO 250
 340 IF Z<>X+Y THEN  PRINT AT 5,0
;"NO, NO, NINNY",,,,"TRY AGAIN"
 350 PRINT
 360 IF Z<>X+Y THEN  GO TO 270
 370 CLS
 380 STOP
 400 CLS : REM MATH DRILL S
 410 REM Subtraction Drill
 420 REM SLOW goes here for 1000
 430 PRINT "SUBTRACTION DRILL"
 440 PRINT
 450 LET X=INT (RND*500)+1
 460 LET Y=INT (RND*500)+1
 465 IF X<Y THEN  GO TO 450
 470 PRINT AT 10,0;X;"  MINUS  ";
Y;"=";"ANSWER?"
 480 INPUT Z
 490 PRINT AT 10,16;Z;"      "
 500 PAUSE 200
 510 CLS
 520 IF Z=X-Y THEN  PRINT AT 5,0;
"RIGHT ON!!  =====  GO ON TO NEXT
"
 530 IF Z=X-Y THEN  GO TO 450
 540 IF Z<>X-Y THEN  PRINT AT 5,0
;"NOT SO FAST! ---- DO IT OVER"
 550 IF Z<>X-Y THEN  GO TO 470
 570 STOP
 750 REM The above program was
    separate programs for 1000
   and combined here. Works on
  2068 also.
 760 STOP
9000 REM To SAVE on the 2068,
       use GOTO 9100
9010 SAVE "math" LINE 9500
9100 REM  To SAVE on 1000/1500
        use GOTO 9000
9110 SAVE "MATH"
9120 RUN
9500 CLS : PRINT "THESE WERE, IN
FACT, THE VERY   FIRST PROGRAMS I
 EVER WROTE ON  MY NEW ZX-81 MANY
 YEARS AGO TO  AID MY YOUNG DAUGH
TERS WITH     THEIR SCHOOLWORK."'
'"WE HAVE COMBINED AND MODIFIED
 THEM A BIT TO ALLOW 2068 USERS
TO RUN THEM."
9505 PRINT ''''"PRESS ANY KEY": P
AUSE 0: CLS
9510 PRINT : PRINT "TO USE EACH S
ECTION, JUST TELL  THE COMPUTER T
O GOTO THE START  LINE OF EACH SE
CTION."
9520 PRINT : PRINT "FOR SUBTRACTI
ON DRILL, ENTER:"''"GOTO 400"''"F
OR ADDITION DRILL, ENTER:"''"GOTO
```

```
 200"''"FOR MULTIPLICATION DRILL,
 ENTER:"''"GOTO 10"
9995 REM The above program was
    separate programs for 1000
   and combined here. Works on
  2068 also.
```

SCROLLING WITH THE T/S 1000

This program scrolls messages of almost any length across the screen.

```
  10 REM A SCROLLING XMAS
     MESSAGE
  15 BORDER 6: PAPER 5: INK 1: C
LS : REM THIS IS FOR THE 2068
  20 LET B=0
  30 LET O=1
  40 LET L=0
  50 LET A$="* * * "
  60 LET A$=A$+"SEASON'S GREETIN
GS FROM THE CLACKAMAS COMPUTER A
PPLIED TRAINING SOCIETY."
  70 LET A$=A$+" MAY YOUR PROGRA
MS ALL BE KEYED IN CORRECTLY AND
 YOUR SCREEN DISPLAYS GLOW WITH
PROGRAMMING WISDOM."
  80 LET A$=A$+" * * *"
  90 LET B$="* * * "
 100 LET B$=B$+"CCATS IS PROUD T
O BE A PART OF THE T/S ACTION. W
E ARE NOW INTO OUR 9TH YEAR AS A
 GROUP AND ARE LOOKING FORWARD "
 110 LET B$=B$+"TO NEW PROGRAMS,
 NEW HARDWARE AND CONTINUED EXPA
NSION OF T/S MAGAZINES. "
 120 LET B$=B$+"WE WANT EVERY SU
PPLIER OF OUR NEEDS TO KNOW THAT
 THEIR EFFORTS TO SATISFY ARE TR
ULY APPRECIATED. * * *"
 130 LET C$="▪"
 140 REM SLOW goes here for 1000
 150 PRINT "CCATS USER GROUP--BU
LLETIN BOARD"
 160 FOR N=0 TO 30
 170 LET C$=C$+"▪"
 180 NEXT N
 190 PRINT AT 5,0;C$;AT 9,0;C$;A
T 6,0;"■";AT 7,0;"■";AT 8,0;"■";
AT 6,31;"■";AT 7,31;"■";AT 8,31;
"■"
 200 PRINT AT 15,0;C$;AT 19,0;C$
;AT 16,0;"■";AT 17,0;"■";AT 18,0
;"■";AT 16,31;"■";AT 17,31;"■";A
T 18,31;"■"
 210 LET C$="
        "
 220 GO SUB 330
 230 LET L=LEN D$
 240 FOR B=1 TO L
 250 IF B>L-27 THEN GO TO 310
 260 PRINT AT A,2;D$(B TO B+27);
 265 PAUSE 7: REM THIS IS FOR
     THE 2068
 270 NEXT B
 280 GO SUB 330
 290 LET L=LEN D$
 300 GO TO 240
 310 PRINT AT A,2;D$(B TO L);
 315 PAUSE 7: REM THIS IS FOR
     THE 2068
 320 GO TO 270
 330 IF O=1 THEN GO TO 380
 340 LET D$=C$+B$+" "
 350 LET A=17
 360 LET O=1
 370 RETURN
 380 LET D$=C$+A$+" "
 390 LET A=7
 400 LET O=0
 410 RETURN
 420 REM
 430 STOP
 500 REM
 510 SAVE "bulitnbrd" LINE 10
```



```
*********************************
*  SEASON'S GREETINGS FROM THE  *
*********************************
```

COMPUTER CHARACTER FORMAT T/S 1000 & 2068

by Dick Wagner

Jack Armstrong's 3 line routine for character generation uses decimal values of binary numbers in the DATA statements. The same system applies to the T/S 1000, but we cannot access generation unless we make up characters in a large 8x8 format such as some programs provide. Now is a good time to look into character format.

By making a diagram, we can arrive at the decimal values of the 8 column by 8 line character format. Zero through 7 lists the columns. The 8 rows must be used from the top down. The DEC column is the decimal values of the 1 & 0 locations across the chart. Input these values into the DATA line for columns 0 through 7 in this order and you will have the / on the v key and on the a key.

To return the a key to normal, either re-power the computer or use Jack's program and define the a by this chart.

We have also produced a simple diagram that gives us one method to easily read the binary number of any row by using the applicable DEC numbers. If rows 1 & 8 are full of 1's, the number is the sum of the DEC column or 255. If rows 2 through 7 have a 1 in columns Ø & 7, the DEC number is 1+128 or 129. Thus we have Jack's DATA line formed which produces an empty square or box. If we change row 4 to 1+8+128, we will have a box with a dot in the middle.

Let your computer compute the DATA input by adding the values by inputting 1+8+128 between ","'s.

7	6	5	4	3	2	1	Ø	DEC.
Ø	Ø	Ø	Ø	Ø	Ø	Ø	1	1
Ø	Ø	Ø	Ø	Ø	Ø	1	Ø	2
Ø	Ø	Ø	Ø	Ø	1	Ø	Ø	4
Ø	Ø	Ø	Ø	1	Ø	Ø	Ø	8
Ø	Ø	Ø	1	Ø	Ø	Ø	Ø	16
Ø	Ø	1	Ø	Ø	Ø	Ø	Ø	32
Ø	1	Ø	Ø	Ø	Ø	Ø	Ø	64
1	Ø	Ø	Ø	Ø	Ø	Ø	Ø	128

A STUDY OF PLOT ON THE 1ØØØ

by Dick Wagner

PART 1

The manual is a good start to look at the PLOT function but you can investigate more about it yourself. If you want to start a PLOT at an exact spot on the screen in relation to a PRINT situation, you must use the correct PLOT coordinates. A PRINT space contains 4 PLOT pixels and the starting point for counting each pixel is the lower left corner of the PRINT space. Try this:

```
  1Ø PLOT Ø,Ø
```

...will give the starting corner of PLOT within the PRINT space. To keep track of our PLOT pixel in space use:

```
  1Ø PRINT AT 2Ø.Ø; CHR$ 8
  2Ø PRINT AT 21,1; CHR$ 8
  3Ø PLOT Ø,Ø
```

Just change line 3Ø to look at 1,Ø; Ø,1;1,1.

You will see that even PLOT numbers for X coordinate, across, is the left half of the PRINT space and the even PLOT numbers for Y coordinates, up, is the lower half of the PRINT space. The first number in PLOT is across and the second is up.

An example may help you:

```
  1Ø PRINT AT 1Ø,1Ø; CHR$ 8
  2Ø FOR N=1Ø TO 3Ø
  3Ø PLOT N,21
  4Ø NEXT N
```

On RUN, the PLOT line will join the bottom of the PRINT character. If 21 is changed to 2Ø, PLOT is a half a line down and for 22, PLOT goes through the bottom half of PRINT and wipes it out.

To put a line on the right side of PRINT, add:

```
  5Ø FOR M=1Ø TO 3Ø
  6Ø PLOT 22,M
  7Ø NEXT M
```

Remember, even X coordinates place the PLOT on the left side of PRINT columns, so, by changing 22 to 19 in place of 18, the PLOT is adjacent to the PRINT character.

A STUDY OF PLOT ON THE 1ØØØ

by Dick Wagner

PART 2

Page 65 of the manual has a statement about combining PRINT and PLOT. If PRINT position moves to the first space after PLOT then the PRINT character is overprinted by PLOT and we have lost it. The exception is if PLOT is moving upward as in my program in the December issue of THE PLOTTER. By using a right-to-left PLOT, we can leave a PRINT character on the screen (lines 20 & 30) and wipe out the PLOT pixel with lines 50 and 60.

To put this to example:

```
  10 PRINT AT 0,10;"PLOT PROGRAM"
  20 FOR N=48 TO 9 STEP -1
  30 PLOT N, 41
  40 PRINT "-"
  50 UNPLOT N,41
  60 NEXT N
```

The 41 places the PLOT in PRINT row 1, as would 40. The bar is actually a PRINT position in the middle of the space. In fact any character or combination (word) you want to display can be PRINTed in this way. The y coordinate (second number) can be even or odd but must correspond to the PRINT row.

Another example is to PLOT a display using a character such as (*).

```
  10 FOR N=60 TO 0 STEP -3
  20 PLOT N,22+ SIN (N/32*PI)
  30 PRINT "*"
  40 UNPLOT N,22+ 20* SIN (N/32*PI)
  50 NEXT N
```

STEP -3 is used to keep from bunching the points at some location. Try -1, -2 & -5 to see the difference. Remember that the points are actually PRINT and not PLOT.

To my knowledge there is no other way to PLOT in BASIC with characters.

A LOOK AT 1 REM (T/S 1000 & ZX81)

by Dick Wagner

Try this test for the address and code numbers in a 1 REM statement. Our first address starts 16514 for the first code number after 1 REM, and then continues for each code number. Randomly input any 10 code numbers between 0 and 255. The corresponding characters will show when RUN and ENTER are pressed. Enter a number each time : cursor shows and on 9/140 press ENTER to see the program with the 10 periods replaced with characters corresponding to the code numbers you input. Check with the character table..

To make your original numbers appear use RUN 200 and they will show along with the addresses. Line 230 acts as a STOP command because code 118 forces the computer to wait for the next line and we have only 1 line.

```
   1 REM..........
 100 LET N=16514
 110 INPUT A
 120 POKE N,A
 130 LET N=N+1
 140 IF N=16523 THEN STOP
 150 GOTO 110
 200 LET N=16514
 210 PRINT N:"    ";PEEK N
 220 LET N=N+1
 230 IF PEEK N=118 THEN STOP
 240 GOTO 210
```

Lines 1 - 150 make a simple loader to put the numbers into line 1. Lines 200 - 240 read line 1 and the address of each code or character.

PERCENTAGE OFF

```
  10 REM PERCENTAGE OFF!
  20 PRINT ,," PERCENTAGE OFF: ";

  30 INPUT "INPUT % OFF: ";P;
  40 PRINT P;" % OFF..."
  50 PRINT '" LEAVES ";(1-.01*P)*
100;" CENTS ON A DOLLAR"
  60 STOP
 100 REM
 110 REM This program will to
        run on the 2068 or the
       1000/1500
 120 SAVE "pctoff": REM 2068 SAVE

 125 STOP
 130 SAVE "PCTOFF": REM 1000 SAVE

 140 RUN
```

SCROLLING BULLETIN BOARDS

by Dick Wagner

Moving message type programs are interesting to people; hence the question: "How do you do that?"

Of course, current TV advertising has us expecting fantastic graphics, but we can program our TS 1000/1500 computers to do interesting message writing.

Horizontal and vertical scrolling small characters is just about our limit. Some variations such as background, framing, flashing, etc. all add interest.

I have several programs based on material from some issues of Synchrosette magazine (now out of business) that should delight the experimenter. These will be published at times in the PLOTTER. Keep them in your program file for reference as you won't find them in books.

The first program will produce large letter words in an upward scroll. Marquee type message presentation (horizontal scroll) will come later. I even have a program that splits the 32 character message in the middle and scrolls right and left at the same time! One program for long messages will encompass some machine language. A loader program plus the MC input, and a means for error correcting will be included. This is not MC programming, so it is really like inputting BASIC that you copy.

The BIG SCROLL program following has a limit of 8 characters per word but will accept many words. It is slow as it shapes each character. The message is continuously repeated but it reforms the letters each time.

```
   5 REM SCROLL MESSAGE WITH BIG
LETTERS
  10 LET S=1
  20 LET G=4
  30 PRINT AT 10,0;"TYPE IN A ME
SSAGE."
  40 INPUT B$
  50 LET B$=b$+"  "
  60 CLS
 100 GO SUB 1000
 110 FOR I=1 TO G
 120 REM CHANGE THIS LINE TO SCROLL
 130 NEXT I
 200 FOR K=1 TO LEN A$
 210 LET C=CODE A$(K)
 220 IF C<128 THEN GOTO 250
 230 LET M$=CHR$ (C-128)
 240 LET C=0
 250 FOR L=0 TO 7
 260 LET P=PEEK (7680+C*8+L)
 270 LET V=128
 280 FOR J=0 TO 7
 290 IF P<V THEN GOTO 330
 300 PLOT 8*(K-1)+J,10-L
 310 GO SUB 2000
 320 LET P=P-V
 330 LET V=V/2
 340 NEXT J
 350 NEXT L
 360 NEXT K
 370 GOTO 100
1000 FOR N=S TO LEN B$
1010 IF B$ (N)=" " THEN GOTO 103
0
1020 NEXT N
1030 LET A$=b$ (S TO N-1)
1040 LET S=N+1
1050 RETURN
2000 IF N<LEN B$-1 THEN LET G=4
2010 IF N>=LEN B$-1 THEN LET S=1
2020 IF N>=LEN B$-1 THEN LET G=2
1
2030 RETURN
9998 SAVE "BIG SCROLL"
9999 RUN
```

Note that the line 9999 will cause the program to AUTORUN when loaded.

CHASE

```
   1 REM CHASE
  10 CLS
  15 LET N=0
  20 LET A=15
  25 LET B=16
  30 LET S=0
  35 LET C=INT (RND*30)+1
  40 LET D=INT (RND*20)+1
  45 LET B$=INKEY$
  50 PRINT AT A,B;CHR$ 133
  55 PRINT AT D,C;CHR$ 137
  60 LET N=N+1
  62 PAUSE 5
  65 IF N=500 THEN  GO TO 200
  70 PRINT AT A,B;"  "
  75 IF B$="5" THEN  LET B=B-1
  80 IF B$="8" THEN  LET B=B+1
  85 IF B$="6" THEN  LET A=A+1
  90 IF B$="7" THEN  LET A=A-1
  95 IF B$="0" AND A=D AND B=C TH
EN  GO TO 500
  97 IF B$="0" THEN  PRINT AT D,C
;"X"
 100 PRINT AT D,C;"  "
 105 IF C=30 THEN  GO TO 35
 110 LET C=C+1
 115 GO TO 45
 200 PRINT AT 10,14;"SCORE IS:";S

 210 STOP
 500 LET S=S+1
 510 PRINT AT D,C;"X"
 520 PRINT AT D,C;" "
 530 GO TO 35
 550 STOP
 600 REM
 610 SAVE "CHASE"
 620 GO TO 1
 640 STOP
 700 RANDOMIZE USR 100: SAVE "cha
se.B1" LINE 1
 710 REM This program designed
    for the 1000, here adapted
   for the 2068
 720 REM Change line 55 to read
    CHR$ 6 for the 1000
 730 REM Delete the pause in
    line 62 for the 1000
```

RENUMBER SUBROUTINE

by: Glen Ten-Eyck

Here is a handy trick for making subroutines that can be renumbered by simple BASIC renumber programs. It does away with GOTO's. At the start of the subroutine, or elswhere in the program if it is more convenient, define a string with a name that is not used elsewhwere or is a throwaway. X$ will be used in the illustration.

```
9000 LET X$="PEEK 23621+256*PEEK
93622"
9010  LET PPC=VAL X$+20: POKE 23
618,(PPC-256*INT (PPC/256)): POK
E 23619,INT (PPC/256): POKE 2362
0,1
9020 PRINT "IT DOES NOT WORK"
9030 PRINT FLASH 1;"IT WORKS!"
```

These are the rules to use -- :

1. Line numbers must advance in a uniform fashion, by 10's is easiest for me.

2. VAL X$ establishes the current line # from the system variables. The amount that is added or subtracted will be the new line number. In the example, add 20 advances the program to 9030 and skips 9020 altogether.

3. 23620 must be POKEd to 1 in order to force the jump to statement 1 or the target line.

Try renumbering this program all that you want. As long as you stay with a spacing of 10, it will work. If you change the line spacing, it will still work if you remember to change the 20 in line 9010 to the appropriate figure (2* line spaces)

I came up with this because the only renumber program that I have does not renumber GOTOs or GOSUBs and everybody that writes a handy subroutine writes it at line 9000 and on.

1000 PROGRAM LOAN STATUS

by Dick Wagner

Try this program to calculate the total interest paid on a loan. Wait for the questions to show after LOADing.

Commonly used formulae are used. See line 270 for the interest paid and line 295 for the unpaid principal balance. I don't have a record of the program source.

Editor's Note: The program does not take into consideration impounds being paid as a part of the monthly payment: taxes, insurance, etc. You would have to figure those separately.

```
   1 REM ************************
  60 GO SUB 1000
 100 PRINT TAB (9);"LOAN STATUS"
 110 PRINT
 120 PRINT ,,,,"THIS PROGRAM WIL
L COMPUTE THE "
 130 PRINT "APPROXIMATE INTEREST
 PAID ON A"
 135 PRINT "LOAN FOR ANY GIVEN P
ERIOD, AND"
 140 PRINT "SUPPLY THE APPROXIMA
TE PRINCIPAL"
 145 PRINT "BALANCE REMAINING."
 146 PRINT ,,,,"HIT ANY KEY TO C
ONTINUE"
 147 PAUSE 40000
 148 CLS
 150 PRINT
 160 PRINT "WHAT WAS THE ORIGINA
L TOTAL      NUMBER OF PAYMENTS?"
 165 INPUT N
 166 PRINT AT 3,25;N
 170 PRINT ,,
 180 PRINT "WHAT IS THE PAYMENT
NUMBER OF   THE FIRST PAYMENT IN
 THE SUBJECTPERIOD?"
 185 INPUT N1
 186 PRINT AT 8,25;N1
 190 LET N1=N1-1
 200 PRINT ,,
 210 PRINT "WHAT IS THE PAYMENT
NUMBER OF   THE LAST PAYMENT IN
THE SUBJECT PERIOD?"
 215 INPUT N2
 216 PRINT AT 13,25;N2
 220 PRINT ,,
 230 PRINT "PLEASE ENTER THE NOR
MAL MONTHLY PAYMENT AMOUNT."
 235 INPUT M
 236 PRINT AT 17,25;M
 240 PRINT
 250 PRINT "PLEASE ENTER THE ANN
UAL PERCENT-AGE RATE."
 255 INPUT R1
 256 PRINT AT 21,25;R1
 257 PAUSE 400
 258 CLS
 260 LET R=R1/1200
 270 PRINT ,,,,
 275 REM FOR TS 1000 COMPUTERS E
NTER SLOW FOR 275
 280 REM FOR TS 1000 COMPUTERS E
NTER THIS FORMULA FOR 280: I=M*(
N2-N1-((((1+R)**(N2-N1))/R)+(((1+
R)**(N1-N))/R))
 290 PRINT "THE TOTAL INTEREST P
AID DURING THE PERIOD IS $";I
 300 REM FOR TS 1000 COMPUTERS E
NTER THIS FORMULA: LET V=M/R)*(1
-(1+R)**(N2-N))
 310 PRINT ,,,,,,
 330 PRINT "THE UNPAID PRINCIPAL
 BALANCE     AFTER PAYMENT NUMBER
";N2
 350 PRINT "IS $";V
 360 PRINT ,,,,
 370 PRINT
 380 PRINT "WOULD YOU LIKE TO SO
LVE ANOTHER PROBLEM?"
 385 INPUT Z$
 390 IF Z$(TO 1)="Y" THEN GO TO
148
 395 CLS
 400 PRINT
 410 PRINT "THAN YOU. I HOPE I H
AVE BEEN OF SOME HELP."
 420 STOP
1100 PAUSE 1200
1105 CLS
1110 RETURN
1120 SAVE "LOAN STATUS"
1130 GO TO 1
```

RIGHT & DECIMAL JUSTIFICATION

by Dennis Jurries

```
  10 LET A$="your number"
    20 PRINT AT x,y-LEN A$;A$
```

x and y are the print row & column

Decimal Justification

```
  10 LET A$="your number"
  20 FOR n=1 TO LEN A$
  30 IF CODE A$(n)=27 THEN GOTO
100
  40 NEXT n
  50 STOP
 100 PRINT AT x,y-n; A$
```

NOTE: In line 30, 27 is the code for the period (.) on the T/S 1000 while it should be 46 on the 2068.

DIM String Decimal Justification..

In dimensioned strings(DIM A$(3,6)), all of the strings have the same length. i.e. 3.2 & 987.15, all have a length of 6. The computer holds the number from left to fight in the string with empty spaces finishing the length. In the above example, "3.2---" and 987.15 are 6 characters in length. ("-"= space). Example:

First we set up the print location:

```
  10 INPUT x
  20 INPUT y
```

Then we initialize the array:

```
  40 LET A$(1)="your #1"
  50 LET A$(2)="  "  #2"
  60 LET A$(3)="  "  #3"
```

Set up the search routine:

```
  70 FOR n=1 TO 3
  80 LET B= VAL A$(n)
  90 LET B$= STR$ B
 100 FOR i=1 TO LEN B$
 110 IF CODE B$(i)=27 THEN GOSUB
1000
 120 IF CODE B$(i)=27 THEN GOTO
140
 130 NEXT i
 140 NEXT n
 150 STOP
```

Print subroutine:

```
1000 PRINT AT x,y-i;B$
1010 LET x=x+1
1020 RETURN
```

NOTE: In above, 27 is the code for the period (.) on the T/S 1000 while it should be 46 on the 2068. "#" is your number of 6 characters or less. Also note that you can set up a DATA table and another LOOP routine to read in the numbers on the 2068. The same is true for the x,y positions if you plan it all out in advance. Then you can print columns of numbers with the periods aligned wherever you want.

For right DIM justification, eliminate all lines from 100 on and put in new line:

```
 100 PRINT AT x,y-LEN B$;B$
```

A QUICKY...OR TWO, OR THREE...

by Dick Wagner

Are you tempted to use UNLESS as a conditional? It would be nice and you can get the same result as:

```
 XXX IF NOT (condition) THEN
(statement)...like this:
```

```
  10 LET C$="y" UNLESS INKEY$="N"
```

so use this replacement form:

```
  10 IF NOT (INKEY$="N") THEN LET
C$="Y"
```

Now how about the modifier UNTIL? Sinclair BASIC doesn't support UNTIL as in:

```
  10 LET Z=Z-21 UNTIL Y<=52
```

then try:

```
  10 LET Z=Z-21
  20 IF NOT (Y<=52) THEN GOTO 10
```

Do you need a FREE for your TS1000? Try this:

9998 PRINT PEEK 16386-16412+256*(PEEK 16387-PEEK 16413).

Editor's note: I think the above should have PEEK as below; it needs to be tested on a TS1ØØØ...

9999 PRINT PEEK 16386-PEEK 16412 +256*(PEEK 16387-PEEK 16413).

How about the program length is bytes for the TS1ØØØ?

```
  1Ø PRINT PEEK 16396+256*PEEK
16397-165Ø9
```

Editor's note: I think this one also needs PEEK:

```
  1Ø PRINT PEEK 16396+256*PEEK
16397-PEEK 165Ø9
```

SCROLLING MESSAGES

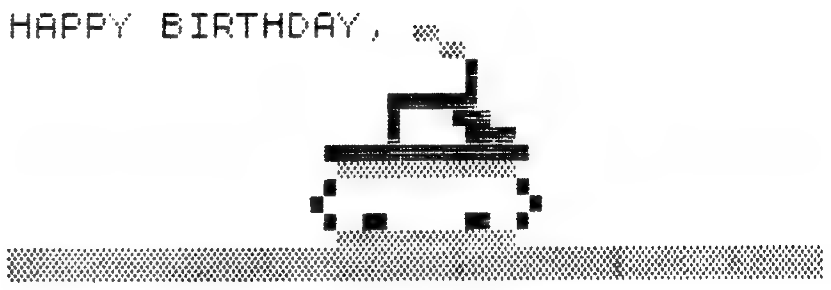
This little graphic program will fit in with graphics Dennis is planning. It uses the same program part for the moving display. My 10 year old grandson gets a kick out of this type of graphics.

The design trick is to place the inverse graphic message just right to make it look like exhaust from the snowcat. It took a little adjusting for the correct "PRINT AT" location so my program should be followed closely.

The REM statements in lines 5 and 6 give the shifted keys to use on the TS 1000 computer in case the final print is not clear.

The snowcat is in fixed position as I have not found a method to make it move.

NOTE: This program was printed with the 2068 computer and a full size Epson printer. Because of the graphics most screen dumps seem to be out of the question. My solution was to use User Defined keys for the graphic blocks not available on the 2068 computer. Next the program was copied as SCREEN$, one screen at a time. Using Jack Dohany's USE9 program for controlled screen dumps, each screen was copied with the printer. The line spacing of 6 lines per inch required a line spacing of 30/216 for my Epson LQ-570 printer in place of 36/214. Probably has something to do with line spacing on the monitor. The 3 sections were then combined into one sheet for cutting and pasting. The graphic image produced by the program was processed the same way.



```
   1 REM I USED A SCROLLING MESS
AGE LIKE "HAPPY BIRTHDAY, JOEL,
HAPPY BIRTHDAY TO YOU ▌". TO MAK
E A GOOD SHOWING TYPE RIGHT TO T
HE END OF EACH LINE, NO EXTRA SP
ACES.
   2 REM LINES 60 THRU 100 PROVI
DES THE SCROLL TO BE ON LINE WIT
H THE SMOKE FROM THE EXHAUST AND
 STARTING AT THE CORRECT POINT.
   5 REM GRAPHIC CHARACTERS ARE
SHIFTED 9 AND SHIFTED NUMBERS/LE
TTERS IN THIS ORDER: D 5 3 5 6 6
 6 5 2 SPACE 6 2 1 G G G G G G G
   6 REM (CONT)T T 2 D G D D D G
 D. IN LINE 55 PRINT 32 DUPLICAT
E CHARACTERS.
   7 REM DICK WAGNER,OCT 1983. S
NOWCAT BY 10 YEAR OLD GRANDSON.
   8 PRINT "INPUT MESSAGE WHEN P
ROMPTED BY ""L"""
   9 PAUSE 1000
  10 CLS
  12 PRINT AT 8,16;"▒"
  15 PRINT AT 9,17;"▀"
  20 PRINT AT 9,18;"▖"
  25 PRINT AT 10,18;"▌"
  30 PRINT AT 10,15;"▄";AT 10,16
;"▄";AT 10,17;"▄";
```

```
  35 PRINT AT 11,15,"▌",AT 11,17
,"▘",AT 11,18,"▛",AT 11,19;"▄"
  40 PRINT AT 12,12;"▘";AT 12,20
;"▌ "
  45 PRINT AT 12,13;"▀▀▀▀▀▀▀";AT
 13,12,"▗▘           ▝▖"
  50 PRINT AT 14,12;"▘";AT 14,13
;"▒";AT 14,14;"▛";AT 14,15;"▒▒▒"
;AT 14,18;"▛";AT 14,19;"▒";AT 14
,20;"▌ "
  55 PRINT AT 15,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒
▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
  60 INPUT A$
  70 LET A$="                    "+A
$
  80 PRINT AT 8,0;A$( TO 16)
  90 LET A$=A$(2 TO LEN A$)+A$(1
)
 100 GO TO 80
```

SAVING RAM MEMORY

by Dennis Jurries

Have you ever written a BASIC program and have used up all of the available memory and are still not finished? This happened to me a while back. I wrote a MacIntosh type display program on Weld Design only to find about three fourths of the way through that I had only 320 bytes left. At this point every time that I would edit a line and try to reinsert it back into the program the computer would delete the line completely. After going back and re- writing the program three times to pick up memory and gaining some, I still could not finish the program. A friend of mine clued me into some memory saving techniques that he had received from another program. The techniques are as follows:

NORMAL USAGE	MEMORY SAVER	BYTES SAVED
1	SGN PI	5
0	NOT PI	5
3	INT PI	5
2,4 & UP	VAL "N"	3

In the last example 2, 4 & up means to use any integer 2 or 4 or on up. When I used these memory savers in my Weld Design program, I found that I had freed up in excess of 12,000 bytes.

ADVICE TS1000

Dick Wagner

This program produces the same text for the TS 1000 that the program "ADVICE 2068" does. The data code is different because of the difference between the 2 computer types. Luckily, the character codes for the TS 1000 are exactly 27 digits less than for the 2068. Comparing code tables, numbers are 20 digits less than the 2068 but none are used here.

The data is in 5 strings, about one for each text line. As the data is in string form, the separating commnas require special treatment as used in the program. This method requires the program to jump to line 200 each time a comma is read.

Line 205 simply converts the text data strings to numeric form so line 210 can print it. To prove this add a PRINT F; immediately after line 205

Considerable testing was required as the program development progressed, but I was working with my 2068 computer because I wished to do the hard copy on my large printer. The data didn't make sense except that spaces were question marks, and periods were 6s. My solution is in the article "READ TS 1000 ON A 2068".

The use of strings to input data, and change the data into numeric digits, is one of several ways of simulating READ & DATA such as is available on a 2068 computer.

FAST puts the program thru nicely but pressing the ENTRY key is required to show the display. Using SLOW displays the characters being printed on the screen. Take your pick by adding a fast or slow line to the program.

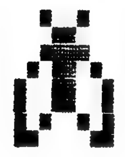
```
   5 REM THIS TS 1000 PROGRAM RE
ADS STRINGS TO PRINT OUT A MESSA
GE
  10 LET A$="50,62,0,46,51,57,42
,55,42,56,57,0,46,56,0,46,51,0,5
7,45,42,0,43,58,57,58,55,42,"
  12 LET B$="0,0,0,0,39,42,40,38
,58,56,42,0,46,0,38,50,0,44,52,4
6,51,44,0,57,52,0,56,53,42,51,41
,0,57,45,42,"
  14 LET C$="0,55,42,56,57,0,52,
43,0,50,62,0,49,46,43,42,0,57,45
,42,55,42,27,"
  16 LET D$="0,0,0,0,0,0,0,0,0,0
,0,40,45,38,55,49,42,56,0,43,27,
0,48,42,57,57,42,55,46,51,44,"
  18 LET E$="0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
,0,0,57,45,42,0,40,45,52,46,40,4
2,0,46,56,0,62,52,58,55,56,27,"
  20 LET A$=A$+B$+C$+D$+E$
 120 LET M=1
 130 FOR N=1 TO LEN A$
 140 IF A$(N)="," THEN GO SUB 20
0
 150 NEXT N
 200 LET F$=A$(M TO N-1)
 205 LET F=VAL F$
 210 PRINT CHR$ F;
 220 LET M=N+1
 230 RETURN
```

SOME T/S -TYPE GRAPHICS

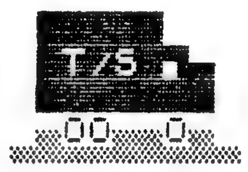
Here are a few T/S 1000-type shapes that might come in handy. They were printed on a large printer using the graphic shapes on 2068 numbered keys plus UD graphics for cross hatching.

FIELD GUN

FLY



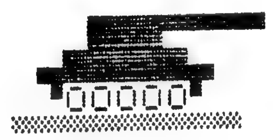
TRUCK



COW BOY

AIR PLANE

TANK #1

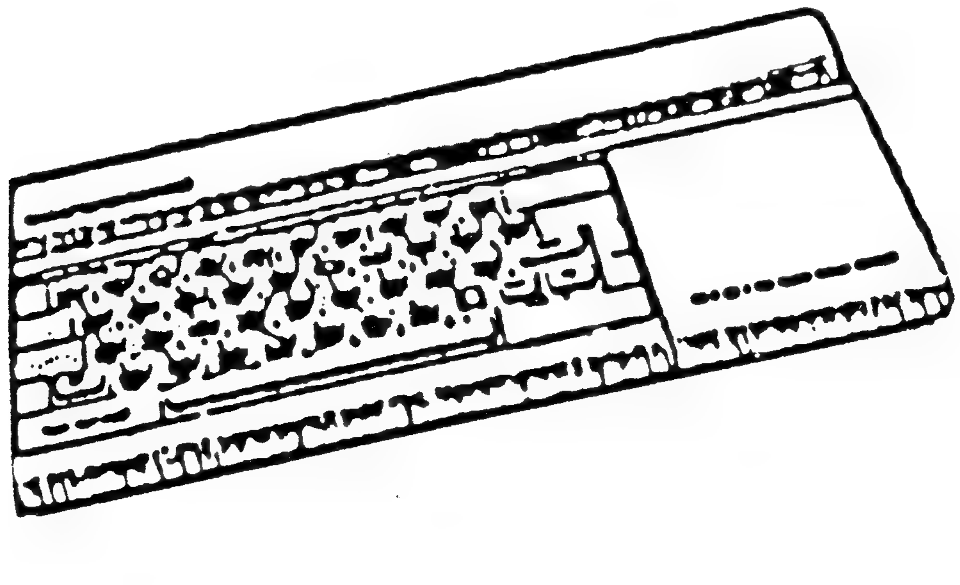


TANK

AUTO 2

Section 2:

T/S2068 SPECTRUM



BOOKWORM
Dick Wagner

This program was developed from a question presented in the Sunday Oregonian (Portland newspaper) of October 25, 1992 by Marilyn in her weekly column of many odities and head scratchers.

The first display of text is the whole definition and question. This was all there was to work with except the answer was given. If it is not the correct answer additional answers may be given. If the answer is correct the player is notified plus the books are displayed on the shelf with the bookworm making its way through the books according to the details given in the beginning.

If the reader desires this to be a self starting program then the SAVE command should end with LINE 50. Put in your own SAVE command based on your storage medium. I named the program "BOOKWORM" because Oliger DOS will accept ten characters in the name.

The DATA defines graphic characters (CAPSHIFT + 9 key) for keys "A", "B", and "C". When RUN is pressed, these keys will change to the graphic characters.

```
  40 REM BOOKWORM PROGRAM
  45 REM DICK WAGNER
  50 PRINT AT 0,0;"There is a se
 t of 5 books on a  shelf. Each b
 ook has 100 pages. A bookworm st
 arts eating on the first page of
  the first book."
  60 PRINT "From there the bookw
 orm eats in a straight line thro
 ugh to the  last page of the las
 t book."
  70 PRINT "Through how many pag
 es did the  bookworm eat?": INPU
 T p: PRINT : PRINT p: PAUSE 50:
 PRINT AT 10,0;"    "
  80 IF P<>302 THEN GO TO 50: IF
  P=302 THEN GO TO 90
  90 CLS : PRINT AT 12,0;"CORREC
 T, 302 PAGES. HERE IS THE  BOOKW
 ORM AT WORK"
 120 FOR n=65368 TO 65368+23
 130 READ d: POKE n,d: NEXT n
 140 DATA 130,254,254,254,254,25
 4,254,254
 150 DATA 254,254,254,254,254,25
 4,254,254
 160 DATA 254,254,254,254,254,25
 4,254,124
 200 PRINT AT 5,5;"A";"A";"A";"A
 ";"A"
 210 PRINT AT 6,5;"B";"B";"B";"B
 ";"B"
 220 PRINT AT 7,5;"C";"C";"C";"C
 ";"C"
 230 PLOT 0,108: DRAW 100,0
 240 PLOT 0,140: DRAW 100,0
 250 PLOT 0,139: DRAW 100,0
 260 PLOT 0,138: DRAW 100,0
 270 PRINT AT 0,0;"Top view of 5
 books"
 280 PRINT "  on book shelf."
 310 FOR X=45 TO (45+27)
 320 PLOT 45,124: DRAW INVERSE 1
 ;(X-45),0
 325 PAUSE 50
 330 NEXT X
```

MORE ON BOOKWORM
Dick Wagner

Our program critic Jack Armstrong finally started up his 2068 and had some fun with my program, BOOKWORM, that was presented in last month's issue of The Plotter. He has been spending his spare tome on a good buy of a MSDOS compatable computer with umpteen programs and other material left in the hard drive by the previous owner.

Having a color monitor for his 2068 he added a color line:

```
45 BORDER: INK : PAPER : CLS
```

He improved the programming with:

```
80 IF p <> 302 THEN CLS: GO TO 50
```

The second trap is not required

He added ends to the bookshelf with:

```
265 PLOT 30,110: DRAW 0,26: PLOT
88,110: DRAW 0,25

290 PRINT AT 3,5;12345
```

DELETE 325 and 330, and add 330, 340,350, and 360.

```
330 PAUSE 15
340 NEXT X
350 INVERSE 0
360 REM  Don't forget to turn  the
INVERSE OFF
```

ADVICE, 2068

Dick Wagner

This program features a bit of wisdom from Charles F. Kettering, long time inventor at General Motors. The character codes in the DATA statements produces the test. Three READ commands are used to format the text.

It should be noted that "n" for each READ loop must equal the length of each statement.

```
  10 REM type this progam and ge
t some sound advice
  20 REM DICK F. WAGNER, NOV. 19
92
 100 FOR n=1 TO 86
 110 READ A: PRINT CHR$ A;
 120 NEXT n
 125 PRINT : PRINT
 130 FOR n=1 TO 26
 140 READ B: PRINT CHR$ B;
 150 NEXT n
 160 PRINT : PRINT : PRINT
 170 FOR n=1 TO 20
 180 READ C: PRINT CHR$ C;
 190 NEXT n
 300 DATA 77,121,32,105,110,116,
101,114,101,115,116,32,105,115,3
2,105,110,32,116,104,101,32,102,
117,116,117,114,101,32,32,32,32
 310 DATA 98,101,99,97,117,115,1
01,32,73,39,109,32,103,111,105,1
10,103,32,116,111,32,115,112,101
,110,100,32,116,104,101,32,32,11
4,101,115,116
 320 DATA 32,111,102,32,109,121,
32,108,105,102,101,32,116,104,10
1,114,101,46
 330 DATA 32,32,32,32,32,32,67,1
04,97,114,108,101,115,32,70,46,3
2,75,101,116,116,101,114,105,110
,103
 340 DATA 84,72,69,32,67,72,79,7
3,67,69,32,73,83,32,89,79,85,82,
83,46
```

WHAT MPH?

Dick Wagner

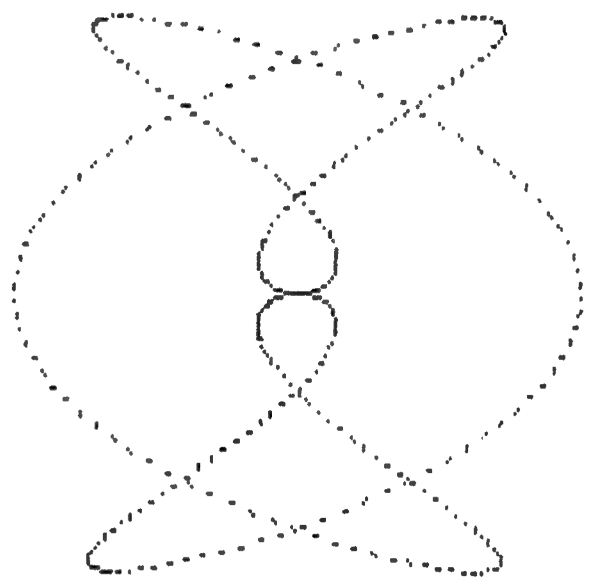
This program will be a head scratcher for some of our readers. Simply, it wants to know how fast you will have to travel to work if 60 MPH puts you there too early and 30 MPH puts you there too late. The one stipulation is that the amount of time you are early is equal to the amount of time you are late.

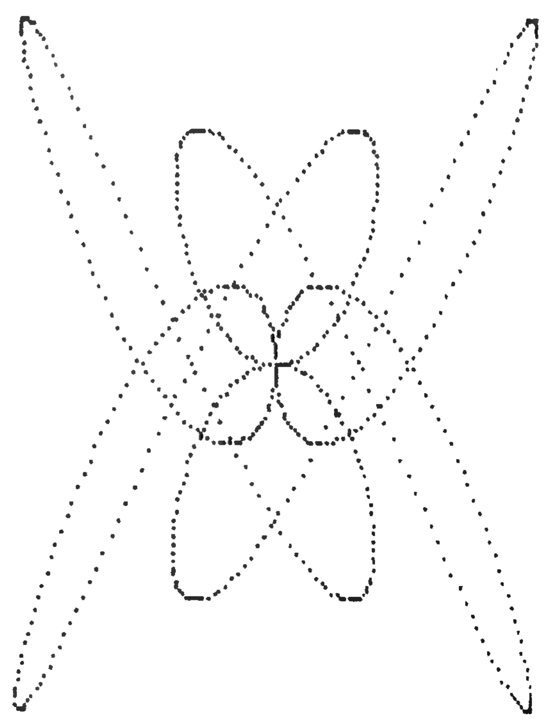
The program keeps repeating until you give the correct answer so use STOP if you give up. You can access the solution by keying in "GOTO 170.

Users working with a TS 1000 will need to change three lines, 110, 120, and 150. THe change is the second CODE which should be "-".

You can assume any distance to travel to work if you so desire.

```
 50 REM Program by Dick Wagner,
 NOV 1992
 60 REM Consider the average sp
eed in MPH for more realistic an
swer, or theoretical constant sp
eed from start to destination.
 80 PRINT "When you go to work
at 60 MPH   you arrive early."
 85 PRINT "When you go to work
at 30 MPH   you arrive late."
 90 PRINT "The amount of time y
ou are      early is also equal
to the      amount of time you a
re late."
 95 PRINT "How fast should you
go to get towork on time? Use av
erage speed or constant speed.":
 PRINT
 100 INPUT "MY GUESS, ESTIMATE,
OR          CALCULATION IS ";MPH
;"MPH"
 110 IF MPH<>INT (CODE "Y"-CODE
"1") THEN PRINT MPH;" MPH IS WRO
NG, TRY AGAIN": GO TO 100
 120 IF MPH=INT (CODE "Y"-CODE "
1") THEN GO TO 150
 150 PRINT "YOU ARE CORRECT WITH
 ";INT (CODE "Y"-CODE "1");" MPH
"
 160 PAUSE 150: CLS
 170 PRINT "DO YOU WANT THE CALC
ULATION FOR THE ANSWER TO BE SHO
WN? Y OR N.": INPUT A$
 180 IF A$="Y" OR A$="y" THEN GO
 TO 200
 190 IF A$="N" OR A$="n" THEN ST
OP
 200 PRINT "DISTANCE = MPH*TIME
WHERE MPH = MILES/HOUR AND TIME
IS IN HOURS"
 210 PRINT "ASSUME A TRAVEL DIST
ANCE OF  60 MILES."
 220 PRINT "FOR 60 MPH, THE TIME
 TO DRIVE IS 1 HR."
 230 PRINT : PRINT "FOR 30 MPH,
THE TIME TO DRIVE IS 2 HR."
 240 PRINT : PRINT "THE DIFFEREN
CE IN TIME IS 1 HR. SO THE NEW T
IME WILL BE 1/2 HR. GREATER FROM
 60 MPH AND 1/2 HR.  LESS FROM 3
0 MPH."
 250 PRINT : PRINT "FOR 60 MPH B
ASE:                MPH=60/(1 HR
+1/2 HR)"
 260 PRINT "THE SOLUTION IS NEW
MPH = 40"
 270 PRINT : PRINT "FOR 30 MPH B
ASE:                MPH=60/(2 HR
-1/2 HR)"
 280 PRINT "THE SOLUTION IS NEW
MPH = 40"
 300 PRINT : PRINT "YOU MIGHT TR
Y OTHER DRIVING      SPEEDS AND T
HE DISTANCE NEED NOTBE COMPARED
TO ONE OF THE SPEEDS, BUT THE SO
LUTION IS EASIER IF YOU DO."
```





INFLATION!!

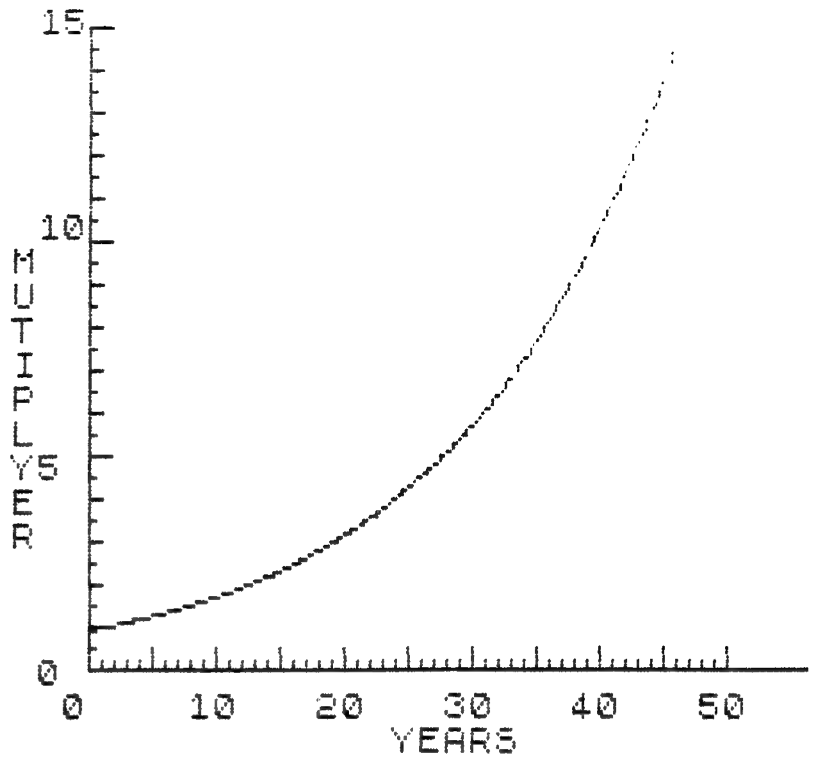
Dick F. Wagner

This short program seems appropriate at this time as there is continued talk of the status of INFLATION. In fact, this subject pops up all too often.

As our readers vary a great deal in age, the information presented will cover an inflation period of 45 years which is about as long as many people remain employed at an income generating job.

This program is based on the formula, M=1.06^n where M is the multiplyer that tell you how many of todays dollars it takes to purchase the same item "n" years from now with an average inflation rate of 6% per annum. Just change the digits after the decimal point to use some other average inflation rate, such as .08 for 8%. The 6% is the average inflation rate for the last 20 years, give or take a little. The plotted curve will go off scale at 6% but it will still plot. For a number output use the formula in direct mode as PRINT (1+ % in decimal)^n for the desired number of years.

I got into this problem while working on a program to help estimate a person's retirement budget and income. It would have been nice to have had such a program available before I retired. Anyway, this inflation multiplyer program is a part of that effort.



```
 10 REM INFLATION CURVE
100 PLOT 25,20: DRAW 0,150
110 PLOT 25,20: DRAW 225,0
120 PRINT AT 21,15;"YEARS"
130 PRINT AT 7,0;"M";AT 8,0;"U"
;AT 9,0;"T";AT 10,0;"I";AT 11,0;
"P";AT 12,0;"L";AT 13,0;"Y";AT 1
4,0;"E";AT 15,0;"R"
140 FOR X=25 TO 228 STEP 4
150 FOR S=20 TO 22
170 PLOT X,S
180 NEXT S: NEXT X
190 FOR X=25 TO 240 STEP 20
200 FOR S=20 TO 25
210 PLOT X,S
220 NEXT S: NEXT X
230 FOR Y=20 TO 170 STEP 10
240 FOR S=25 TO 29
250 PLOT S,Y
260 NEXT S: NEXT Y
270 FOR Y=20 TO 170 STEP 5
280 FOR S=25 TO 27
290 PLOT S,Y
300 NEXT S: NEXT Y
310 FOR Y=20 TO 170 STEP 50
320 FOR S=25 TO 32
330 PLOT S,Y: NEXT S: NEXT Y
340 PRINT AT 0,1;"15"
350 PRINT AT 6,1;"10"
360 PRINT AT 13,1;"5"
370 PRINT AT 19,1;"0"
380 FOR N=0 TO 25 STEP 5
390 PRINT AT 20,N+2;N*2
400 NEXT N
410 FOR N=0 TO 46 STEP .2
420 PLOT 25+4*N,19+10.14*(1.06^
N)
430 NEXT N
```

MORE ON TASWORD 2

by: Dick Wagner

At one time or another, users of TW2 have had occasion to look at or change printer codes. The menu gives this option, the TW2 programmers recognizing the needs of diffeent printers. Perhaps the reader has changed printers, thus requiring one or more changes.

I have altered codes several times, starting with a Olivetti printer. For sure I didn't understand all that I was doing. Why did some codes in the original list include 32? Nothing was said in the meager instruction manual about this. Lucky for me, my changes didn't seem to effect the printer output, if I knew enough to look for them.

Recently I came across a letter by P. F. Green (Rotterdam) in the ZX Computing Monthly for February 1987, in the Cross Wires column in which he addressed the subject of code 32. From his discussion it appears that the code 32 is used to make the lines justify properly on the printed page.

Refering to the 2068 list of characters and codes in that manual, it will be noted that 32 is the code for "space" (we all new that, didn't we?) Where the right justify does not print properly, look at the codes being sent to the printer. If only default codes are being used, then look at these to see if the right justify can be adjusted by adding a space in one of those codes. It the reader has added a printer code to the text, say at one point where Italics is desired, then this could be the culprit.

A good test would be to delete a 32 in a code and use the new one to print a text. This should show the effect of the code 32 when right justiying.

Another TW2 improvement was published in the March-April 1985 issue of Syncware News, page 5. This was written by Duncab Teague. (It is surprising how much information can be dug out of these old magazines.)

Several times in our old THE PLOTTER issues we have published information on how to revise the first "HELP" page of TW2. This information is a little different approach that will make it possible to edit both "HELP" pages. Why change them? Have you changed printer codes for a printer function that was not on the tabulation of codes? Have you changed printers but have left the headings unchanged? Have you made a note of the correct function as a substitute to keep track? Why not correct these changes properly and have a real nice menu?

Thisprocess takes the "HELP" pages out of the memory locations and puts them into text files so they can be edited. Then the edited files are returned to their original addresses.

#1--with TW2 loaded, STOP to get to basic and then type in the command mode (no line numbers) the following:

```
 FOR i=0  TO  1535:  POKE
(33280+i), PEEK (5478+i): NEXT i
```

#2--GO TO 20 and select "y" to return to the text file. the first of the "HELP" pages in now in the text file and can be read and revised.

#3--After editing, reverse the addresses and poke the screen back to memory.

#4--The second "HELP" pages in sent to the textfile with the following line:

```
 FOR i= TO 1535: POKE (33280+i),
PEEK (56320+i): NEXT i
```

NOTE: this address is 1536 higher.

#5--Repeat as in #2 and #3. Beverse the addresses and poke the screen backto the new memory location.

Now you have "HELP" pages to suit your changes and needs. SO easy!

No Doubt that all who have been using TW2 with the AERCO type of parallel interface have these pokes:

57999	127
58001	103
58008	127
58014	219
68015	127

Note that 127 is the port address.

Special printer symbols with SYMBOL SHIFT in EXTENDED MODE:

Key	Symbol	Instead of
y	[	
u	]	
p	@	copyright
a	~	slanted quotes
s	|	solid bar
d	\	
f	{	
g	}	

Special printer symbols with SYMBOL SHIFT in the normal mode:

Key	Symbol	Instead of
h	^	up arrow
x	#	pound sign

Note: as this was written on MSCRIPT (my WP preference), 2 symbols were printed via imbeded printer codes. The back slice is used for end of line or paragraph so could not be inserted in the text. The symbol for "d" is on the "s" key. This does not print in the text.

MORE TASWORD II

by: Dick Wagner

Some of us have used a program for some time, obtained more/newer equipment, and then find that we have to adapt that old familiat program to use it. This has happened to many of us who have used TASWORD II back in the early tape drive days.

There have been articles and word of mouth explanations on fitting this progran to different printer interfaces. In another issue I will guide you thru the process of changing the HELP page to match those changed printer codes. printer name, etc. Why keep a crib sheet of your changes when you can correct the menu?

Here ia a review of several popular interfaces. POKE the code nembers into the corresponding addresses, either in direct mode or use POKE program. BUT first, run an address/code check with a loop program to see it there are some that do not need changing. Maybe just a few POKEs will do it.

	AERCO IF	A & J	OLIGER/ HACKSEL
ADDRESS	CODE	CODE	CODE
57999	127	65	127
58000	203	230	203
58001	103	4	103
58005	0	211	0
58006	0	66	0
58007	211	62	211
58008	127	4	127
58009	0	211	0
58010	62	65	62
58011	247	175	247
58013	251	65	251
58014	219	201	219
58015	NC	NC	127

DISK MANAGER

by: Rod Gowen

The Disk Manager Program was written by Larry Kenny and Jack Dohany. This set of instructions will serve for the first version which was included on all LKDOS master disks.

To LOAD: simply RANDOMIZE USR 100: LOAD "DSKMAN.B1"

It comes up running. It will automatically do a "CAT" of tis disk or whatever disk is in the drive taht you run it from.

At this point, yoiu will see a losting of the files and you will see an information line as well as a menu of functions at the bottom of the screen.

The first item is "cat:0-3(0)". This tells you that you are looking at a catalog of drive (0). To CATALOG a new drive, just press a number (0-3), (0-4_ if you have ramdisk), and you will then be prompted to enter a "filter". That is, if you only wish tosee basic files, you could enter ".B^"and only basic files would be shown on the screen. Play with this, you willsoon get the habg of it.

The next item is "cursor:5-8". This tells you that you can use the 5/6/7/8 keys to move the high-lighter bar around the screen with complete wrap-around. NO NEED TO USE SHIFT KEYS! Just press the number for the direction you wish to move.
At eh right of this line is "mark:X/I". This tells you that you can use one of these two keys to mark items. If you press the "X" key you will see an "X" in inverse video appear to the left of the item that is currently highlighted. If you press the "I" key, you the "X" appear to the left of EVERY item on the screen! (NOTE: One small exception to this: if there is an AUTOSTART on the disk, the auto marking will stop as soon as it hits it! I do not know why, but I assume that this is a "bug" and no one has yet fixed it. You can manually mark all other items on the screen and they will be acted upon.) It you press the "I" again all of the marks will be removed. Or, if you press "X" while on a marked item, the mark will be removed.

FUNCTIONS:

On the lower line of the screen you will see the function. This is just as it sounds. If you wish to RENAME an item, just highlight it and press"R" and you will be prompted to enter the information needed to give the file a new or changed name.

The ERASE function is much the same. Just highlight the item(s) that you wish to ERASE of delete from the disk and press the "E" key. You will have a chance to change your mind on almost all of these functions. The prompt will give you a "Y/N" option. If you accidently erase a file and would like to get it back, you will have to use one of the other available file recovery program. Some of these will be appearing in future issues of RGM CATALOG UPDATES.
The MOVE function is just that; it will allow you to MOVE one or more files from one drive to another or to the same drive. If you are using a single drive system, you can copy a file or an entire disk as long as you do not mind swapping disks a few times. When you press "M" you will beasked to tell the computer what drive you want to move the file(s) to. Then it will display the settings and ask if you wish to continue.

The BACKUP function is a bit deciving. It is actually just a SAVE routine for the DISK MANAGER program itself. All you need to do

If you want to add this program to a set of utilities on your boot disk of utility disk, you can modify line 76 to send you back to your main menu bu simple taking out the STOP command and replacing it with: RANDOMIZE USR 100: GO TO D: RANODMIZE USR 100: NEW tl go back to your original menu, provided it is and AUTOSTART. It it is not, you can use: RANDOMIZE USR 100 GO TO d: RANDOMIZE USE 100: LOAD "MENU.B1". The "d" is the drive # you wish to use.

IN CONCLUSION:

I hope that you get a lot out of our little endeavor, Please feel free to call or write regarding this publication or any of the products/materials listed herein. We would be glad to hear from you.

If you wish to print these pages, you can do so. Simply use you version of the MC for Tasword (unless you have an A&J printer i/f, in which case you can do it with this version) and load and print whichever pages you wish. We have set up the pages to run a close as possible to 66 lines as possible.
is to CAT the drive you want to copy or save the program to and then press "B".

Finally you have the QUIT function. Not much I can tell you here! If you press "Q", you will see a STOP error line appear at the botton of the screen and you will be in basic.

ADDITIONAL NOTES:

After much experimentation and trial and error, I have found a couple of nice things to know about this program. I want to pass these notes on to you so that you will not have to do as I did. These are small but critical items to the operation of this program.

I have many disks that have a LOT more files than would fit on one screen full of 2 columns and was rather put out that there did not seem to be any way that I could access and work with any but the last 30 or so files on a disk. I saw "scroll?" appear at the botton of the screen and upon pressing ANY key, I saw the filenames scrolling by and I was presented with the menu ONLY when it reached the last of the files on the disk catalog. WELL: it was really very simple: SIMPLY PRESS THE "BEACK" KEY! BUT BE CAREFULL! PRESS <<ONLY>> THE BREAK KEY OND NO OTHER KEY OR IT WILL CONTINUE TO SCROOL! As soon as you press BREAK, you will be presented with the menu at the botton of the screen, If you finish working with the files on-screen, you simply press the number of the drive that you are working with and re-CATALOG the disk, Press ENTER to get past the first screen full and then press BREAK again to work with the next screen. You just continue to do this until you are finished with that disk. I have a couple of Quad disks with over 140 files and it works fine on them.

As you can see, there are three files connected with this program. They are DSKMAN.B1, Move.C1 and cat.CR. If you use the built-in BACKUP, you will automatically saveall three parts. Otherwise, you must be sure to copy all three parts to the disk you are going to use.

AUTO-RUN PROGRAMS

by Dick Wagner

Auto-run programs can be stopped after LOADing by using this procedure. ENTER MERGE "" (only on the 2068) and then ENTER LOAD "". You won't see LOAD "", but it is there. Start your recorder and stop on O.K. On ENTER the LISTing will come up. It will also be easier to BREAK a running program.

WAFER TIPS

PRO/FILE 2068

FOR THE A&J

This month I will explain the four lines that must be altered and show you the actual altered lines before and after to help you make the change-over yourself in order to make PROFILE 2068 run on the A&j MICRODRIVE. As I said there are only four lines to change, two in the LOADER program and two in the primary program.

You must first start the LOADing process by LOAD "". Then, as soon as you see the screen appear that says the program is LOADing, please wait, HIT BREAK and STOP THE TAPE. Press the ENTER key and you will see the LOADER PROGRAM on the screen. It will look like the one below. Study #1 and #2. Change the lines to read exactly like #2. As soon as this is done, put a new FORMATTED WAFER in your drive and SAVE the LOADER PROGRAM by using the direct command: "save "@1,pf"LINE 1. After SAVEing the LOADER, reset the 2068 (turn it off and back on), and rewind the tape and again LOAD the program, this time do not stop it. As soon as the program is LOADED, you will see the prompt asking if you want to LOAD or CREATE a file. At this time, you will press DELETE to get rid of the left Quote and then press STOP and ENTER. This will STOP the program so that you can make the changes.

After you have make the changes to the lines as shown below, you will SAVE the revised program by using GOTO 8000.

There you have it! A microdrive operating version of PROFILE 2068.

To SAVE files from the program, just remember to use the microdrive command (i.e.-"@1,file--name") when the prompt asks you for a name for the file that you wish to SAVE.

IMPORTANT NOTE: IF YOU CAN'T READ ANYTHING WHEN YOU STOP THE PROGRAM OR THE LOADER, IT IS BECAUSE THE PAPER AND THE INK ARE THE SAME COLOR. YOU MUST TYPE IN THE COMMAND: "INK 7" AND PRESS ENTER TWICE. THEN YOU CAN READ THE LISTING. IF NOT, THEN TYPE IN THE COMMAND "PAPER 0" AND PRESS ENTER. You can use any other combination that has a contrast from one to the other.

LISTING #1

```
  1 BORDER 0
  2 PAPER 0
 10 CLEAR 63487
 30 PRINT AT 5,8; PAPER 1; INK 
7;"* PRO/FILE 2068 *";
 40 PRINT AT 7,4; INK 7;"c 1984
 By THOMAS B. WOODS";AT 10,11; I
NK 6;"P.O. Box 64";AT 11,7; INK 
6;"Jefferson, NH 03583";AT 19,7;
 PAPER 1; INK 6; FLASH 1;"LOADIN
G"; FLASH 0; INK 6; PAPER 0;" Pl
ease wait";at 0,0; LOAD "p/f"COD
E 63488,2046
 50 LOAD "pro/file"
```

LISTING #2

```
  1 BORDER 0
  2 PAPER 0
 10 CLEAR 63487
 30 PRINT AT 5,8; PAPER 1; INK 
7;"* PRO/FILE 2068 *";
 40 PRINT AT 7,4; INK 7;"c 1984
 By THOMAS B. WOODS";AT 10,11; I
NK 6;"P.O. Box 64";AT 11,7; INK 
6;"Jefferson, NH 03583";AT 19,7;
 PAPER 1; INK 6; FLASH 1;"LOADIN
G"; FLASH 0; INK 6; PAPER 0;" Pl
ease wait";at 0,0; LOAD "@p/f"CO
DE 63488,2046
 50 LOAD "@pro"
```

```
8000 SAVE "@2,p/f"CODE 63488,204
6
8010 SAVE "@,pro" LINE 9995
```

PRIMARY PROGRAM #1

```
8000 SAVE "p/f"CODE 63488,2046
8010 SAVE "pro/file" LINE 9995
```

PRIMARY PROGRAM #2

BUZZ SAW

This is another 2068 graphis to make a picture of "WAGNER'S BUZZ SAW" and work work with angles. If the reader will recall, when you use angles they must be in radians for computer formulas. Simply, one degree of a circle is equal to PI/180, or 0.017453 radians. In the following program line 110 changes the degrees defined in line 100 to radians. Line 110 changes the 15 degree steps required to produce 24 saw teeth to radians. Line 130 draws the cutting edge of the teeht to a length of 12. Line 140 plots the tips of the teeth so DRAW can be used in line 150 to draw the back edge of the teeth.

Two circles are drawn for the hole to give "thickness" to the blade. This corresponds to the "depth" produced by line 30.

If this is printed on a printer just follow the explanation given in the "snail" program. When you finish it should look like the image shown.

```
 10 PLOT 40,9: DRAW 174,0: DRAW
0,158: DRAW -174,0: DRAW 0,-158
 20 PRINT AT 19,8;"WAGNER'S BUZ
Z SAW"
 30 PLOT 41,167: DRAW 174,0: DR
AW 0,-158
100 FOR A=0 TO 360 STEP 15
110 LET B=A*0.017453
120 PLOT 127+62*SIN B,87+62*COS
B
130 DRAW 12*SIN B,12*COS B
140 PLOT 127+62*SIN B,87+62*COS
B
150 DRAW -18*SIN (B+.7),-18*COS
(B+.7)
160 CIRCLE 127,87,10
170 CIRCLE 128,88,10
180 NEXT A
```

RANDOM NUMBERS ON THE 2068

by: Dick Wagner

Most computers using some form of BASIC language provide the ability to generate random numbers. Readers who have used random numbers in programs recognize that random numbers are usually in intergers (whole numbers), but the program may not require it. This is because the random number generator produces a range of numbers of numbers between 0 and something less than 1 (a very small something). The 2068 computer has a number generator that produces 873 numbers from 0 to 872, all between >0 and <1.

The numbers generated by RND are always the same set of 873 numbers. RANDOMIZE n starts the next RND number. If the reader types 10 print RND and runs it, the same number will be displayed each time. The trick is to start the generator RANDIMIZE n with any number greater than 0 and less than 65537. The reason that n is greater than 0 is that 0 is reserved for a special seeding process.

For a test, key in this program

```
10 RANDOMIZE N
20 PRINT RND
```

and run it with different n values. Try it several times with the same value of n and the repeat of each random number will be obvious. Now try n=872 and a number close to 0.99905396 is displayed. This is the largest number I could find. Next try

n=873 and the display will be almost 0, · my reading was 0.00018320547. These are close to the limits for the generator, it just recycles with larger numbers up to 65536 and the same range of 873 numbers are available.
There is a logical sequence to the numbers generated for RND. The equation is: RND=(75*(n+1)-)/65536 where n ranges from 0 to 872. All numbers produce are less than 1.

When an interger is used by keying INT, the number is a whole number by rounding down. Thus adding 1 will bring it back to the required number. An example is where 6 is required as a result of RND calculation. So INT(RND*6)+1 will give a number of 6 because the largest number, 0.9990536 multiplies by 6 will give 5.9943273 when rounded down and 1 added equals INT 6. The smallest RND number that will give 6 this way is about 0.83311462. Thus there are a range of numbers that will give INT 6. In fact for this example there are about 45 numbers in the generator that will! I have found that for some calculations RND is not always duplicated exactly. In this example RANDOMIZE 728 and then PRINT RND*6 should give 5.0055542. This is because 728 is the smallest number that will seed RND and produce INT 6 in this manner. This was easy to find by trial and error using the equation for RND previously given.

Should the reader wish to interigate all of the numbers available for seeding RND use this program:

```
  10 FOR n=1 TO 872
  20 PRINT (75*(n-1)+1)/65536
  30 NEXT n
```

The reader has probably noticed that RANDOMIZE 0 has not been addressed. RANDOMIZE 0 is a special case so it is not to be used for starting RND in the ordinary way. RANDOMIZE 0 will force the generator into a very acceptable set of random numbers that will not be duplicated except under very rare chance. The computer keeps track of the number or video frames sent out up to 6537 since start up, or from a POKE 23670,0 and POKE 23671,0 to reset, or from starting over after reaching 65537. A video frame takes 0.16 milliseconds, or 0.00016 seconds.

How about a simple program that flips a coin for HEADS or TAILS? This should be a good example of random if enough tests are run.

```
 20 RANDOMIZE 0
 30 FOR N=1 TO 200
 40 LET r=RND
 50 IF r<.5 THEN LET h=h+1
 60 IF r=>.5 THEN LET t=t+t
 70 PRINT A 0.5;"HEADS";" ";h
 80 PRINT AT 0, 20;"TAILS";" ";t
 100 PAUSE 20
 110 NEXT n
```

This program self runs for 200 loops, posting the number of times HEADS and TAILS have come up lines 50 and 60 set up the range of numbers produced by RND. This does not require an integer number.

A fair consistency of times for HEADS and TAILS will be shown because of the small test base of 200 there will be variations. Now that the explanation of RANDOMIZE , and RND has been given, the readers should be able to produce a program for a guessing game. Guess what the computer has flipped, HEADS or TAILS, and keep a tally of the wins and losses. Send a copy of your program to me at our mailing address and maybe it will get into out newsletter.

MORE 2068 GRAPHICS

by: Dick Wagner

The first program draws a "snail" by processing a program for a spiral of 360 degrees. It also draws straight lines between the center and the points, forming the shell. The original spiral program is line 20. The radial lines are produced by line 30. Try it first without line 30 to see how the spiral is formed. By adding line 30 the program draws each line in order.

If this image is dumped to a large printer, the correct ratio of width to length is accomplished by a fudge factor to compensate for printing too wide by the ratio of 72:60. Just insert 60/72 in the SIN function of lines 20 and 30 as (60/73)*(15*v) *SIN v. If you want proof of this, produce a "circle" with this line and send it to the printer.

```
 10 FOR X=0 TO 2*PI STEP
PI/150:PLOT 127+50*SIN
X,87+50*COSX :NEXT X
```

Now put the factor in a (60/72)*50*SIN x, etc and run again.

This comes about because COPY prints with 60 dots per inch in the lines and 72 dots for line spacing. A printer driver that makes screen dumps can probably be corrected, if required, by inserting the proper code for the dot density.

The 2040 printer requires a correction factor of 1:24 as it prints elongated.

```
 5 REM a graphic snail
 7 REM by Dick Wagner
 10 For V=0TO 2*PI STEP PI/38
 20 PLOT 127+(15*V)*SIN V,
 60+(15*V)*COS V
 30 PLOT 127,60: DRAW (15*V)*SIN
 V,(15*V)*COS V
 40 NEXT V
```

```
   5 REM a graphic snail
   7 REM by Dick Wagner
  10 FOR v=0 TO 2*PI STEP PI/38
  20 PLOT 127+(15*v)*SIN v,60+(15
*v)*COS v
  30 PLOT 127,60: DRAW (15*v)*SIN
 v,(15*v)*COS v
  40 NEXT v
9998 STOP
9999 RANDOMIZE USR 100: SAVE "sna
il.B1" LINE 1
```

FABRIC CONVERSION CHART

by: Dick F. Wagner

Convert One Width/ Length to another Width/Length

Sewing patterns call for a given amount of fabric yardage based on a pattern size. The width is not always available for a desired fabric type and/or design. This conversion chart (from a fabric manufacturer) gives the conversion of width and length to another length for a desired width. Compensation for difficult design match should be considered in the called for width so conversion includes the added amount.

Print the title, FABRIC CONVERSION CHART, for the first line with 5 spaces each side. This is underlined 33 spaces in pica.

The remainder of the chart is printed in compressed mode. Set your printer for compressed, or insert the code in the test.

The top line is an underline 62 characters long. The next line just prints in the word "INCHES"

The 3rd line prints "WIDTH", and at the following TABs, TAB 8+2 spaces, "32"; TAB 15, "36-38"; TAB 22+2 spaces, "39"; TAB 29+2 spaces, "41"; TAB 43, "44-45"; TAB 50, "52-54"; TAB 57, "56-60". This line is Underlined.

The 5th line starts with "Yards", then at each TAB key in the line of numbers. For the remaining lines of numbers. For the remaining lines just key in the numbers.

INCHES - WIDTH	32	36-38	39	41	44-45	50	52-54	56-60
Yards	1 7/8	1 3/4	1 1/2	1 1/2	1 3/8	1 1/4	1 1/8	1
	2 1/4	2	1 3/4	1 3/4	1 5/8	1 1/2	1 3/8	1 1/4
	2 1/2	2 1/4	2	2	1 3/4	1 5/8	1 1/2	1 3/8
	2 3/4	2 1/2	2 1/4	2 1/4	2 1/8	1 3/4	1 3/4	1 5/8
	3 1/8	2 7/8	2 1/2	2 1/2	2 1/4	2	1 7/8	1 3/4
	3 3/8	3 1/8	2 3/4	2 3/4	2 1/2	2 1/4	2	1 7/8
	3 3/4	3 3/8	3	2 7/8	2 3/4	2 3/8	2 1/4	2
	4	3 3/4	3 1/4	3 1/8	2 7/8	2 5/8	2 3/8	2 3/8
	4 3/8	4 1/4		3 1/2	3 1/8	2 3/4	2 5/8	2 3/8
	4 5/8	4 1/2	3 3/4	3 5/8	3 3/8	3	2 3/4	2 5/8
	5	4 3/4	4	3 7/8	3 5/8	3 1/4	2 7/8	2 3/4
	5 1/4	5	4 1/4	4 1/8	3 7/8	3 3/8	3 1/8	2 7/8

GETTING THE IN ON THE 2Ø68...

by Dennis Jurries

The number of possible I/O ports on the 2Ø68 is 65536. These are used by the computer for communicating with things like the keyboard or the printer, and they can be controlled from BASIC by using the IN statement which is a function like PEEK...

```
  IN address
```

The keyboard is divided up into 8 half rows of 5 keys each.:

```
  IN 65278 reads CAPS SHIFT to V
  IN 65Ø22 reads A to G
  IN 6451Ø reads Q to T
  IN 63486 reads 1 to 5
  IN 61438 reads Ø to 6
  IN 57342 reads P to 7
  IN 4915Ø reads ENTER to H
  IN 32766 reads SPACE to B
```

The value of these addresses when no key is depressed is 31. Starting from the outside of the keyboard and moving towards the center, the values are 3Ø, 29, 27, 23, & 15. On some models of the Spectrum, NO KEY DEPRESSED is 255 and the other values are 254, 251, 247 & 239.

The following program illustrates this:

```
  10 FOR N=Ø TO 7:REM HALF ROW #
  20 LET A=254+*(255-2^N)
  30 PRINT AT Ø,Ø; IN A: GOTO 3Ø
```

When you finish with each half row, press BREAK and then type NEXT N.

The IN function can be used in many ways in a program. One way is to use it to change a position of something on the screen. Say the variable X has some value; LET X=X-(IN65Ø22=3Ø) will decrease the value of X by 3Ø only if the A key is depressed.

TIPS ON USE OF COLOR COMMANDS IN EXTENDED MODE

by Jack Armstrong

(Editor's Note:- See article Page 48)

When you want to put a color command into a line so that it will be printed on screen in a color, you do not have to use the INK command.

Instead, you can imbed the color command in the line itself by using the CAPS/SHIFT & the SYMBOL/SHIFT together and enter the EXTENDED MODE. Then hold down the CAPS/SHIFT & press the key representing the color of your choice.

This will imbed the color command just in front of the character that is the 1st one to print in the color. Then continue typing the characters until you are ready to change back to BLACK. Then go into the extended mode again and, while holding down the CAPS/SHIFT key again, press the BLACK key. This returns you to the ordinary way of black letters again.

This is an interesting way to set off your REM statements so that they are eye-grabbers on your screen.

There are ways to change the PAPER commands also, but I haven't been able to figure all of them out yet. I'm trying to see if the FLASH command can be inserted, too.

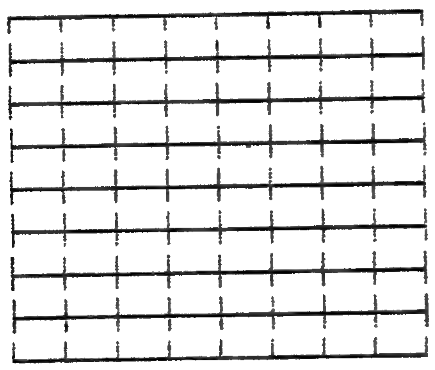
Be sure to turn off the command at the end of the section you want to have in color. If you don't, the command will affect all characters following. It must be turned off. No space will be taken for the command in the line, but you will see the cursor change color and everything that you type will be affected.

USER DEFINED GRAPHIC CHARACTERS

by Dennis Jurries

According to Chap. 18, page 163 of the TS 2068 Manual, you can create and use 21 user defined graphics. Some people may not be aware that you can make far more than 21. That chapter describes how you can use BIN numbers to create those graphics by POKING them into the addresses reserved for graphic characters. This article will show how to create your graphics easier and faster than the manual shows and how to do more than 21.

The first step is to either get some graph paper or create your own by making up a grid 8 boxes wide and 8 boxes deep:



...write numbers over the columns from left to right: 128,64,32,16,8, 4,2,1...this is the 2's compliment representation of the columns as described in a previous article about "ATTRIBUTES". You should refer to the manual page 164.

Next draw in your character as shown in the manual filling in the small boxes that your lines pass through. The next step is to write the total of the column #'s at the right of that row. As an example, the 1/4 symbol used in the manual is:

```
   row 1=64        =64    THESE ARE
   row 2=64+4      =68    DECIMAL
   row 3=64+8      =72    NUMBERS
   row 4=64+16     =80    INSTEAD
   row 5=32+8+2    =42    OF USING
   row 6=64+8+2    =74    BINary
   row 7=8+4+2+1   =15    as in the
   row 8=2         = 2    manual.
```

Use the following program to insert your character into one of the 21 user defined graphics character keys A through U.

```
1000 FOR A=USER "e" TO USR "e"+7
1010 READ b:POKE a,b
1020 NEXT A
1030 DATA 64,68,72,80,42,74,15,2
```

The above 21 user defined graphics characters are, as are the standard character set for the computer set up and controlled by a ROM routine. In order to use more than those 21, you will have to bypass the ROM routine and, instead, go into RAM.

According to the manual, address 23606 always contains the address that is 256 less than the address of the character set. In fact, this is the least significant byte of the address, less 256 with address 23607 containing the most significant byte of the total address. If you PEEK these addresses you will get 0 and 60 respectively. This corresponds to address 15360 in ROM; add 256 to it giving 15616 as the starting address of the standard character set.

If you use this method, you must retain in RAM any character that you still want to keep...from the space to the copyright symbol as the characters are called up in order. The following is a sample method for replacing the letters both capital and lower case, while retaining all other characters. This means you will be redesigning the alphabet if you want a new Pont in your 2068.

Lower RAMTOP to protect the new set:

```
  10 CLEAR 64499
  20 LET a=PEEK 23606+256*PEEK 2
3207+256
```

Take 33 characters from ROM & put in RAM:

```
  30 FOR n=0 TO 32
  40 FOR i=0 TO 7
  50 POKE 64500+i+8*n, PEEK (a+i
+8*n)
  60 NEXT i
  70 NEXT n
```

Read in characters to replace 26 capital letters:

```
  80 FOR n=33 TO 58
  90 FOR i=0 TO 7
 100 READ X
 110 POKE 64500+i+8*n,x
 120 NEXT i
 130 NEXT n
```

Take 6 characters from ROM & put in RAM:

```
 140 FOR n=59 TO 64
 150 POKE 64500+i+8*n, PEEK (a+i
+8*n)
 160 NEXT i
 170 NEXT n
```

Read in characters to replace 26 lower case letters:

```
 200 FOR n=65 TO 90
 210 FOR i=0 TO 7
 220 READ x
 230 POKE 64500+i+8*n,x
 240 NEXT i
 250 NEXT n
```

Take last 5 characters from ROM & put in RAM:

```
 260 FOR n=91 TO 95
 270 FOR i=0 TO 7
 280 POKE 64500+i+8*n, PEEK (a+i
+8*n)
 290 NEXT i
 300 NEXT n
6000 DATA............ETC.
```

With the 21 UDG's defined and the character set redefined, you now have 118 user defined characters. To complete the program, fill in the DATA statements with your decimal numbers.

To use these characters, after running the program: POKE 23605,244 POKE 23607,250. To revert back to the standard character set: POKE 23606,0 POKE 23607,60.

QUICK UDG PROGRAM

by Jack Armstrong

For all of the T/S 2068 users who want a better way to use the capabilities of the computer to enhance their games or even to dress up their more serious programs, here is a bay to define their graphics without resorting to using the BIN codes suggested in the manual. That method requires inputting 64 numbers for each character & this method only needs 8. You do need to have a graph grid to design the characters and get the decimal numbers to use in your DATA statements.

This little routine will POKE numbers into locations in RAM which will give you user-defined graphics on lower case keys in the graphics mode--you use all the letters from a through u.

```
6000 FOR a=USR "a" TO USR "a" +7
6010 READ UDG: POKE a, UDG: NEXT a
7000 DATA 255,129,129,129,129,12
9,129,255
```

These DATA numbers will put a box on the a key in graphics mode.

ENTERING AND RECALLING IN MACHINE CODE

by Dennis Jurries

This subroutine will allow the user to enter up to 277 characters into machine code and recover then at will. The routine consists of four parts. The first part is the machine code loader program.

```
  10 FOR X=55501 TO 55533
  20 READ Z: POKE X,Z
  30 NEXT X
  40 DATA 62,0,205,48,18,225,70,
35,229,33,183,215,126,35,254,128
,56,250,16,248,126,35,254,128,20
0,215,24,248,201,205,205,216,1,2
01
```

RUN the program above, then DELETE if. The second part consists of six lines that allow the text you input into machine code to be recovered.

```
  10 FOR X=1 TO Z
  20 POKE 55533,X
  30 RANDOMIZE USR 55530
  40 PAUSE 0
  50 NEXT X
  60 STOP
```

The third part of the routine allows the user to put text into machine code. Once the user puts the text in and tests it then this part may be deleted.

```
 100 LET Z=1: LET J=55524: LET A
$=""
 110 PRINT  "Input text (limit 32
characters   if you want to print
text on the  24th line only).  Pr
ess ENTER    when finished with t
ext."
 120 PRINT
 130 GO SUB 300
 140 INPUT A$
 150 IF LEN A$>=55500-J THEN PRI
NT AT 0,0:"Text too long.  Input
text.": CLS : LET J=J-LEN A$: LE
T Z=Z-1: GOTO 110
 160 FOR I=1 TO LEN A$
 170 POKE J,CODE A$(I)
 180 IF J<55500 AND I=LEN A$ THE
N POKE J+1,128
 190 IF J<55500 AND I=LEN A$ THE
N LET J=J+1
 200 IF J,55500 THEN LET J=J+1
 210 NEXT I
 220 LET Z=Z+1
 230 GOSUB 400
 240 LET Z=Z+1
 250 STOP
 300 PRINT "Text #";z;'"    ";277
-LEN A$;" CHARACTERS REMAINING."
 310 RETURN
 400 PRINT
 410 PRINT "More text? (y/n)"
 420 PAUSE 0
 430 IF INKEY$="y" OR INKEY$="Y"
THEN CLS : GOTO 110
 440 RETURN
```

The above is set up to print on the 24th line. Several different messages can be run if you add a line, 45 CLS. If you wish to print text from the top of the page then POKE 55502,2. To change back to the 24th line POKE 55502,0.

The following is the OP CODE for the machine code routine entered in part one above. This is the fourth part of the sub-routine.

```
  55501 LD A,0
  55503 CALL 4656
  55506 POP HL
  55507 LD B,(HL)
  55508 INC HL
  55509 PUSH HL
  55510 LD HL,55223
  55513 LD A,(HL)
  55514 INC HL
  55515 CP 128
  55517 JR C,55513
  55519 DJNZ B,55513
  55521 LD A,(HL)
  55522 INC HL
  55523 CP 128
  55525 RET Z
  55526 RST 16
  55527 JR 55521
  55529 RET
  55530 CALL 55501
  55533 01
  55534 RET
```

BREAKING & SAVING 2068 PROGRAMS

by Dennis Jurries

Most programs that you can purchase for the 2068 start running when loading and if you try breaking into them they dump or lock up the computer or will not accept the break command. These programs usually consist of at least three programs in one. The first part consists of a simple loader that tells the computer to load the SCREEN$ and load the machine code program and may have some basic program that intermixes with the machine code program. The first part may be hidden when you break into it because the BORDER, PAPER and INK colors all the same. Change the paper color if this is so and re-LIST it.

The following procedure works in all cases:

```
MC       - machine code
           program name
BASIC    - BASIC program name
SCREEN$  - SCREEN$ program name
```

1. LOAD the program until the screen display is displayed, then BREAK and LIST. This listing will give you the starting address of the machine code (found in the line: RAND USR #), The name of the MC, and SCREEN$ programs.

2. LOAD the MC program (LOAD "MC" CODE)

3. PEEK the MC program looking for the length. This can be done by looking in the MC until you find a long group of 0's (say, about 32).

4. SAVE "BASIC" LINE 1 SAVE "Screen" SCREEN$ SAVE "MC" CODE #1, #2 #1 is the starting address from RAND USR. #2 is the length of the MC routine determined by subtracting the starting address from the ending address found when you peeked the MC and found the start of the 0's or by using the top of memory 65535 (i.e., say start address from RAND USR 42000: 65535-42000 = 23535, then SAVE "MC"CODE 42000, 23535). 5. After completing the above and VERIFYing each step, press NEW and ENTER. LOAD the program copy you just made and try it out.

FANCY FONTS

Have you ever purchased a program that used a character set other than the one that came with your 2068? Ever wished you could duplicate that character set in your programs? Here's how you can do it!

First, there is a system variable, called CHARS, that contains the location of the currently active character set. This is really the address of the character set, less 256, to allow the computer to find the correct character information.

The address of CHARS is 23606 and 23607. CHARS is set to contain 15360 upon turning on your computer. You can verify this by typing:

```
PRINT PEEK 23606+PEEK 23607*256
```

You fill find that 23606 contains 0 and 23607 contains 60. This is how we tell the computer where to find the character set. So, we can use the above statement to find the fancy character set we are interested in.

Now, load in your program with the fancy characters. BREAK into BASIC and type the above statement. Add 256 to answer you just got and write it down. This is the start of the fancy character set.

The length of the character table is 768. This is arrived at by knowing that there are 96 definable characters, each needing 8 bytes of information to define their shape.

We now have all the information we need to "lift" the character set from the program, for our own use. Type in the proper command to save the character information to whatever storage medium you use. The tape command is:

```
    SAVE "name"CODE start#,768
```

Remember to use the first number we calculated, in place of the word "start#", in the above command.

The only thing we need to do now, in order to use our fancy characters is to load them and POKE the system variable, CHARS.

This is easiest to accomplish if we calculate a start address for our character table, such that it starts on a new page of memory. This allows us to change character sets with a single POKE.

The highest address we can use, without interfering with the UDG'S, is 64512. 64512/256=252. We need to remember to subtract 256 before our POKE. The easiest way to do this is to subtract 1 from 252.

Now, type in the needed commands to load your fancy character set and POKE the system variable. The tape commands are:

```
  LOAD "name"CODE 64512
  POKE 23607,251
```

Of course, you could load your character set to almost any address. You need only adjust the address in the LOAD command and the value POKEd to CHARS, in the above commands.

You can now switch between the character sets at will. All that is needed is to POKE 23607,60 for the normal characters or POKE 23607,251 for the fancy characters.

If you are still using the 2040 printer, this can be used to make the printing more legible. I use a "fat" character set, for bold printing, to my 2040. Have Fun!

TRANSPARENT INK!

by Dick Wagner

Please refer to Syd Wyncoop's article on Color Codes in the January 1987 issue and my article in the December 1986 issue (More on Attributes).

I recently read more on this subject that sheds additional light on the use of color attributes. This was found in "The Art of Programming The ZX Spectrum" by M. James, an English publication. There will be a review on this small book later.

It is possible to assign an attribute to a specific location on the screen and call it up at will. It can be a color as PAPER or INK, and it can be BRIGHT and or FLASH. Try these two lines:

```
 10 PRINT AT 10,10: FLASH 1;"*"
```

Run and there is a flashing *. Now add this line and RUN:

```
20 PRINT AT 10,10; FLASH 8;"BANG"
```

The B is the only character flashing because it is taking the action given to this location when FLASH 8 is used.

In a program structure it might be used like this:

```
  10 PRINT AT 10,16; FLASH 1;"**
****1;"****** **" 80 PRINT AT 10
,0; FLASH 8;"This li ne has a fl
ashing word!"
```

The word "flashing" replaces the 8 *'s by use of FLASH 8. The same principle applies to BRIGHT, INK and PAPER and in combination.

For PAPER 8 any new character at the designated location will assume the color of the paper background assigned by PAPER. Such characteristics can be called "transparent" in the sense that it lets the original attribute show through.

This ability to assign a feature to a specific location would seem to have value in games in conjunction with a color monitor, as well as with a B & W monitor.

MUSIC WITH BEEP
by Dick Wagner

It is easy to determine the pitch of a musical not with BEEP generated by our 2068 computers. Based on the standard note A (440 Hz), middle C is 261.624 Hz. (See table page 187-189 in 2068 manual).

The pitch, based on middle C, is calculated by the computer with the formula f=2 p/12 * fc, where fc is the frequency of middle C, p is the pitch and f is the frequency in Hertz of the note. This formula is changed to solve for pitch, p: p=12*LN(f/261.624)/LN 2, where LN is the natural logarithm (base e) which is shifted key Z, f is the note frequency in Hz. Don't be alarmed by logarithms because the computer knows all about them. Use this program to print pitch values of desired notes:

```
 10 INPUT "Frequency in Hz ";f
 20 LET P=12*LN (F/261.624)/LN 2
 30 PRINT f;" ";Pitch value= ";p
 40 GOTO 10
```

From the table, select the frequency of some notes. Example:

```
261.624 (p=0) octave #1
523.248 (p=12) octave #2
1046.496 (p=24) octave #3
2092.992 (p=36) octave #4
4185.984 (p=48) octave #5
```

(p is progressing 12 notes each time). Try other combinations such as D (293.664) and continue the same way. You will see that the pitch is in steps of +12 each time.

Now determine the relationship between whole notes (not sharp or flat). C through B will be in the order of 1,2,4,7,9,11 & 12 (close). The missing numbers are the semi-tones 3,5,6,8 & 10 and they are the sharps/flats. Calculations for notes below middle C will have minus values.

The above relationship is a music standard, I think it is now clear about programming BEEP as far as music is concerned--the designers based the tone on the standard musical scale with BEEP t,p as a note of duration t and frequency corresponding to the pitch where pitch = 0 (261.624 Hz). This also explains the minus values used for p.

Note duration is important in music. The use of click in most computer programming is of short duration and is not musical. When composing on the computer, make a table of the notes (as letters) and the corresponding pitches for reference. Also assign duration numbers for proposed timing, such as whole note (T=2), 1/2 note (T/2=1), 1/4 note (T/4=1//2), 1/8 note (T/8=1/2) & 1/16 note (T/16=1/8). This puts all of the note durations on terms of T so if it is desired to redefine T then all durations are changed automatically.

The commonly used POKE 23609,n varies the note duration so this POKE won't work in music. The reader might check out my findings about the pitch of this note, I decided that BEEP t,34.51 seemed to duplicate the pitch. Use a very short duration to match. I changed the formula to f=261.624 * ^(P/12) and calculated f = 1920.39 which is about B in octave #3.

Fractional values of duration and pitch work with BEEP. The definition for BEEP x,y is x in seconds and y in pitch semi-tones above or below middle C and my experiments bear this out.

The note duration is mainly by trial as too short a duration is just a click. Low pitched notes must have time for at least a full excursion. A 30 cycle sound is 30 cycles per second which is 0.0333 seconds. This short a period is not suitable for pleasing sounds.

As you experiment don't expect too much from the tiny speaker. Try elevating the computer a little to let the sound out.

TRANSFER FILES WITHOUT RETYPING - PROFILE 2068

by Rod Gowen

If you, like me, purchased a new copy of PROFILE 2068 with all of the modifications already in it, you have found yourself with the task of retyping everything in your files into the new copy...NOT NECESSARY!

The book is not too clear on how to transfer files, so her is how to do it. These are direct from the author, Thomas B. Woods, with many thanks.

FIRST PHASE: Load your original file, the one you have filled, and do the following:

Break into BASIC by DELETEing the left quotes and ENTERing STOP.

With BASIC LISTing on the screen, enter PRINT p. This will tell you how many bytes of date are in d$, which is your file. Write this number down.

Then type in this line:

```
PRINT PEEK 23627+256*PEEK 23628
```

Add 6 to this number and write it down. This number is the starting ADDRESS of your files. with this information you can now SAVE your files as raw BYTES by using the following direct command:

```
SAVE "FN"CODE AAAAA,BBBB
```

"FN" stands for FILENAME, the AAAAA stands for the ADDRESS, the BBBB stands for the number of BYTES, all of which you found by the method above.

Once you have SAVEd the raw BYTES you can now reset the 2068 by turning it off and back on.

SECOND PHASE: Now LOAD the new version of PROFILE 2068 into the 2068 and proceed as follows:

Go to BASIC as you did before and repeat the procedure to find the starting ADDRESS of your file area in memory. It will be a different number than with the first file. Write this number down for use in LOADing your files. That line again was:

```
PRINT PEEK 23627+256*PEEK 23628
```

As before, add 6 to the number and you have the starting ADDRESS of the area to which you want to put yOur files. Use the following line to LOAD you file DATA into the new version:

```
LOAD "FN"CODE AAAAA,BBBB
```

Again, "FN" stands for FILENAME, the AAAAA stands for the ADDRESS, the BBBB stands for the number of BYTES that you SAVEd from the first PROFILE.

After your DATA is LOADed, you must enter the following line:

LET P=BBBB

This will reset your file size to accept the DATA. Once that is done, type in GOTO 1, and, if you have done it right, your files will be safely in the new version of PROFILE 2068. Just look at how much typing you have saved!

My Favorite Triangle

by Dick Wagner

Here is a program short enough to make it fun. Sorry, it only works on the 2068. It is easy to follow through but just try putting it together the first time. Your eyes get kind of crossed.

Maybe you can find some diagrams like this that make good viewing and furnish them to the PLOTTER.

The text part is printed on a Olivetti Ink Jet printer, using the A&J Micro Drive interface and Tasword 2 word processor. I am curious about how it will reproduce in the PLOTTER. The program part is on the 2040 printer.

```
   5 PRINT "      My Favorite Tri
angle                   D.F.W."
  20 PLOT 48,88: DRAW 0,-12: DRA
W 96,-48
  30 DRAW 12,6: DRAW 0,96: DRAW
-12,6: DRAW -96,-48
  40 DRAW 84,-42: DRAW 0,60
  50 PLOT 144,28: DRAW 0,84
  60 PLOT 156,130: DRAW -84,-42:
DRAW 60,-30
  70 PLOT 144,112: DRAW -60,-30
```

SEE PAGE 74 FOR ILLUSTRATION

BANK STATEMENT CHECKER

by: Jack Armstrong
Revised by: Dick Wagner

This program has been in our newsletter material file for a long time. It will work for the TS 1000 and 2068 computers. A neat way to balance that bank statement. Just key in the number of checks, and then amounts on the prompts.

THE EDITOR

THIS ROUTINE WILL PICK UP DOLLAR AMOUNTS AND SUBSCRIPT THEM IN THE FRIST LOOP AND THEN IN THE SECOND LOOP IT WILL ADD THEM UP. THIS GIVES YOU A VERY POWERFUL TOOL I PROGTAMMING.

JACK ARMSTROMG'S PROGRAM WITH VARIATIONS AND REVISIONS BY DICK WAGNER.

```
   1 REM by Jack Armstrong
   2 REM rev. by Dick Wagner
  10 LET ct=0
  20 PRINT : PRINT "HOW MANY CHEC
KS? ";
  30 INPUT n
  40 PRINT n
  50 DIM a(n)
  55 FOR k=1 TO n
  60 PRINT : PRINT "INPUT AMOUNT
OF EACH CHECK"
  70 INPUT a(k)
  80 PRINT a(k)
  90 NEXT k
  95 PRINT
 100 FOR j=1 TO n
 110 LET ct=ct+a(j)
 115 PRINT a(j)
 120 NEXT j
 125 PRINT "________"
 130 PRINT ct;" THE TOTAL OF ";n;
" CHECKS."
 200 STOP
9999 RANDOMIZE USR 100: SAVE "ban
k.B1" LINE 1
```

USING POINT

by: Dick Wagner

What is POINT? Our 2068 manual simply states that POINT tells you whether a PLOT point specified by coordinates is PAPER color (0), or INK color (1). The coordinates for x range from 0 to 255 and y ranges from 0 to 175. Note: input other than PLOT also in recognized, but the results are by coordinates. I recall trying this in the early days of 2068 experiments, but it didn't seem of much value.

Sharon Aker's book, T/S 2068 Basics and BEYOND, actually puts POINT to some use with this short program:

PROGRAM #1

```
 10 INPUT "Press any key ";a$
 15 PRINT a$
 20 PRINT AT 10,0
 25 FOR y=175 TO 168 STEP-1
 30 FOR x=0 TO 8
 35 PRINT POINT (x,y);
 40 NEXT x: PRINT: NEXT y
```

Just input any character including graphic and see the result.

For some time I have thought that there should be some way to print the pixel addresses or coordinates of a graphic figure. For practical purposes it should be a small figure as an 8x8 figure requires 128 nunbers for 64 addresses if the 8x8 is all INK color. Many designs may be much less than this. If such could be accomplished, then it would be possible to provide a design to any 2068 user. So, a design equal to the space of about 4 characters should be practical to convert to addresses.

Make these changes in lines 35 and 40 to get a numerical print out of a character for input:

```
 35 IF POINT (x,y) THEN PRINT x;
"";y
 40 NEXT x: NEXT y
```

Now you have actual coordinates printed for the shape you INPUT. We are not interested in printing addresses of PAPER color. The first column is the x line and the second is the y column. Obviously the amount of information that can be displayed in this manner is very limited. See the 2068 manual, page 152 for a display of coordinates used on the 2068. If the whole screen was printed out in this manner the data printed would be enormous because the POINT for each part of data would also be printed! The solution is to do this on paper.

A more useful record can be obtained by sending the data to a 2040 printer. In this way the image can be any place on the screen, Make these line changes:

```
 33 IF POINT (x,y) THEN LPRINT
x;",";
 35 IF POINT (x,y) THEN LPRINT
y;"";
```

This gives a line print out with a comma joining two coordinate numbers and a space between addresses. Try it with RUN and see the difference. This is more like a DATA line.

The reason that all of this works is that "IF POINT (x,y)" means "ON" if it is a true statement. The next step is to work this DATA into a useable statement. This requires the numbers to be copied into DATA lines, adding commas at the spaces between addresses.

For example, a Volkswagen BUG was used as it was 2 characters wide, being formed from a user graphic assigned to keys A and B. The basic design again was from Sharon Aker. Later, some changes were made to include parts of the next two lower character positions plus a slightly longer figure in these two character positions, Thus the figure was 9 pixels high and 20 pixels long. These changes will be decribed later, as well as reshaping the front end of the car. All easily accomplished! The original design can be obtained from page 116 for those with access to this book.

The next step was to develop a program to print out the coordinates for every INK 1 for this figure. At the same time a base line (road) was added. The program follows with line 20 being the LOAD command for Oliger. Just change this line to fit the readers equipment. Line 55 adds the ability to print on all 23 lines and scroll properly. Line 25 adds the base line under the BUG. You will have to wait until you get a SCREEN$ of BUG before you can use this next program. I will give you the data soon for this.

PROGRAM #2

```
 10 LET a$="BUG"
 20 LOAD /a$ SCREEN$
 25 PLOT 0,167: DRAW 19,0
 30 PRINT AT 1,0
 50 FOR y=175 TO 165 STEP-1
 55 POKE 23692,255
 60 FOR x=0 TO 20
 70 IF POINT (x,y) THEN PRINT x;
",";y;" ";
 80 NEXT x: NEXT y
```

The raw DATA can be used to reproduce an inage of the BUG by copying it as lines of DATA, adding commas in the spaces, However, we do not have a SCREEN$ yet to load in for line 10. The next program with DATA will be easier to use and to change. The program will reproduce the BUG and after a SCREEN$ save, called BUG, can be used in the above program to see the DATA that was used in the following program. As the reader does not have the image untilthis last step, the SCREEN$ can be used in this way.

PROGRAM #3

```
 100 FOR a=1 TO 125
 110 LET m=175
 112 LET n=174
 114 LET o=173
 116 LET p=172
 118 LET q=171
 120 LET r=170
 122 LET s=169
 124 LET t=168
 126 LET u=167
 130 READ x
 135 READ y
 140 DATA 5,m,6,m,7,m,8,m,9,m,10
,m
 145 DATA 3,n,4,n,5,n,6,n,7,n,8,
n,9,n,10,n,11,n,12,n,13,n
 150 DATA 2,o,3,o,4,o,5,o,6,o,7,
o,8,o,9,o,10,o,11,o,12,o,13,o,14
,o
 155 DATA 1,p,2,p,3,p,4,p,5,p,6,
p,7,p,8,p,9,p,10,p,11,p,12,p,13,
p,14,p,15,p
 160 DATA 0,q,1,q,2,q,3,q,4,q,5,
q,6,q,7,q,8,q,9,q,10,q,11,q,12,q
,13,q,14,q,15,q,
 165 DATA  0,r,1,r,2,r,3,r,4,r,5
,r,6,r,7,r,8,r,9,r,10,r,11,r,12,
r,13,r,14,r,15,r,
 175 DATA 2,s,3,s,4,s,5,s,6,s,7,
s,8,s,9,s,10,s,11,s,12,s,13,s
 180 DATA 3,t,4,t,11,t,12,t
 185 DATA 0,u,1,u,2,u,3,u,4,u,5,
u,6,u,7,u,8,u,9,u,10,u,11,u,12,u
,13,14,u,15,u,16,u,17,u,18,u,1
9,u,20,u
 200 PLOT x,y
 210 NEXT a
```

Keying in this program and data will produce an image of a VW BUG at the upper left corner of the screen. To show the convenience of this last program to modify a figure, change lines 140 and delete 5,m; 145 and delete 3,n,4,n,; 150 and delete 2,o,3,o,; and 160 with delete 1, p,. Now RUN again and see a better shaped BUG. SAVE this SCREEN$ if you like it. The SCREEN$ of BUG can now be used in program #2 as if you had it originally.

A next step might be to look at the new expanded image with the first program. However, this image takes up 3 character spaces across the screen (x axis) and 2 character spaces up (y axis). This program must be adjusted to match the new image.

You might have a preferredscreen location for the figure. Just take the last program and chonge the PLOT by subtracting from the y coordinate the start of the new location. And if moved across the screen, add the change in the z coordinates. Suppose the location is 50 points down and 50 points across. Line 130 READ x+50 and line 135 READ y-50 are the changes. RUN and SAVE again. Also, be sure and save the progran each time.

Assume you don'thave the screen location of a figure on the screen. Just make a broad bracket of the location. It might look like it is in the upper half of the screen and somewhere across! Change line 50 as 50 FOR y=175 TO (166-85) STEP-1, and line 60 as 60 FOR 0 TO 255. Now run this change and allof the INK 1 pixels will be located.

Several images can be displayed on the screen and saved back as a new SCREEN$ by using the following program that duplicates and image across the screen, or even in various locations. The duplicating program is very simple and works like a charm. The program starts with a VW BUG developed previously and saved. Then it is combined into 2 VW BUGS. Line 210 provides for 9 pixel rows and line 220 produces a figure 39 pixels long including the original figure, 19 being added to the first 20.

PROGRAM #4

```
 200 LOAD /"BUG"SCREEN$
 210 FOR y=175 TO 175-8 STEP-1
 220 FOR x=0 TO 19
 230 IF POINT (x,y) THEN PLOT
x+20,y
 240 NEXT x: PRINT: NEXT y
 250 SAVE "BUGS"SCREEN$
```

Save this program and the SCREEN$ of the 2 BUGS. Now that there are 2 BUGS we will join them into a total of 12 BUGS. adding 10. The following program will demonstrate how to nake a border of "Volkswagen Bugs On Parade".

PROGRAM #5

```
 110 LOAD "BUGS"SCREEN$
 120 REM this image is from
program#4
 130 FOR Y=175 TO 175-8 STEP-1
 140 IF POINT (x,y) THEN PLOT x+
20,y
 150 IF POINT (x,y) THEN PLOT x+
40,y
 160 IF POINT (x,y) THEN PLOT x+
60,y
 170 IF POINT (x,y) THEN PLOT x+
80,y
 180 IF POINT (x,y) THEN PLOT x+
100,y
 190 NEXT x: PRINT: NEXT y
 200 PRINT AT 2,0;"--Volkswagen
Bugs On Parade--"
```

+ + + + + + + + + + + + + + + + +

```
    5 REM UNION JACK Designed
      1/85 by Jack Armstrong
   10 LET a=0: BORDER 5: PAPER 2:
 INK 1: CLS
   20 PLOT 0,0: DRAW 0,175: DRAW 2
55,0: DRAW 0,-175: DRAW -255,0
   30 FOR a=120 TO 136: PLOT a,174
 : DRAW 0,-174: NEXT a
   40 FOR a=80 TO 96: PLOT 0,a: DR
AW 254,0: NEXT a
   50 LET b=239: LET c=-159
   60 FOR a=1 TO 16: PLOT 0,ABS c:
  DRAW b,c: LET b=b+1: LET c=c-1:
 NEXT a
   70 LET b=-174: LET c=254
   80 FOR a=1 TO 16: PLOT a,174: D
RAW c,b: LET b=b+1: LET c=c-1: NE
XT a
   90 LET b=158: LET c=238
  100 FOR a=16 TO 0 STEP -1: PLOT
0,a: DRAW c,b: LET b=b+1: LET c=c
+1: NEXT a
  110 LET b=254: LET c=174
  120 FOR a=1 TO 16: PLOT a,0: DRA
W b,c: LET b=b-1: LET c=c-1: NEXT
  a
  130 PAUSE 90: PRINT #1;AT 1,0;"
      GOD SAVE THE QUEEN!"
  140 PAUSE 0: PAPER 7: CLS
  150 STOP
  200 REM
  210 RANDOMIZE USR 100: SAVE "uni
onj.B1" LINE 10
```

DAY OF THE WEEK

by: Dick Wagner

Here is a 2068 program that will provide an answer to that unanswered question, on what day of the week were you born?

It is a good group or party program that allows people to input their birth date and out comes the day of the week. Of course, it can be used for any date to find out what day it falls on.

The only input to watch out for is the month first (1-12), then the day (1-31), and then the year (xxxx). Note that the year must be later than 1752.

```
 90 REM Program to calculate   t
he day of the week
 110 DIM d$(7,9)
 120 LET d$(1)="Monday"
 130 LET d$(2)="Tuesday"
 140 LET d$(3)="Wednesday"
 150 LET d$(4)="Thursday"
 160 LET d$(5)="Friday"
 170 LET d$(6)="Saturday"
 180 LET d$(7)="Sunday"
 190 PRINT "Date (MM,DD,YYYY)?";
 200 INPUT m,d,y
 205 IF d<=0 THEN  GO TO 640
 210 GO SUB 500
 220 PRINT d$(z)
 230 GO TO 190
 500 IF y<=1752 THEN  GO TO 620
 510 LET n=INT (.6+1/m)
 520 LET L=y-n
 530 LET p=m+12*n
 540 LET c=L/100
 550 LET y1=INT (c)
 560 LET z1=INT (c/4)
 570 LET z3=INT (5*L/4)
 580 LET z4=INT (13*(p+1)/5)
 590 LET z=z4+z3-y1+z1+d+5
 600 LET z=z-(7*INT (z/7))+1
 610 RETURN
 620 PRINT "The year must be afte
r 1752"
 630 GO TO 190
9998 STOP
9999 RANDOMIZE USR 100: SAVE "day
.B1" LINE 1
```

CHINESE FORTUNE CALENDAR

Dick F. Wagner

Several years age my wife and I were touring in Nova Scotia Province, Canada. We spent one night in the town of Yarmouth, which is on the western tip of Nova Scotia. Yarmouth is the terminal for ferries from Portland and Bar Harbor, Maine. For a change we elected to eat at a Chinese restaurant. Our table place mats carried an interesting Chinese Fortune Calendar. Thinking it would offer an interesting computer program opportunity, I requested a clean one from our waitress. The following program is from this place mat. Any errors cannot be blamed on food spots.

The format of the calendar is 12 messages or fortunes. Each message is given the name of an animal. The messages indicate which animals are most compatable with that particular birth year. There are usually 2 or 3 animals in this catagory. As the Chinese Zodiac is in 12 year cycles, a series of years, 12 years apart, are associated with each animal. Thus years 1912, 1924, 1936, etc. are associated with the animal Rat. The animals named in order of years are Cock, Dog, Boar, Rat, Ox, Tiger, Hare, Dragon, Snake, Horse, Sheep, and Monkey (the way I figure it Year 1 seems to be Cock).

The following text was given to explain the use: "The Chinese Zodiac consists of a 12 cycle, each year of which is named after a different animal that imparts distinct characteristics to its year. Many Chinese believe that the year of a person's birth is the primary factor in determining that person's personality traits, physical and mental attributes and degree of success and happiness throughout his lifetime. To learn about your Animal Sign, find the year of your birth among the 12 signs running around the border."

This program produces the appropriate animal and message when you input your birth year. You cannot put all messages on the screen due to space limitations. To read each one it is necessary to input all 12 years, one at a time. One way is to start with the year such as 1921 and then progress one year at a time for 12 years. The screen can then be copied each time to get a printed display.

Note that each animal message ends with a compatible list of other animals. If the messages are hard copied you can pick out the favorable animals, otherwise it is necessary to read in on the screen.

This is interesting to use for seeing how compatible your spouse is, your parents, your children, etc. Try it at a gathering of friends. Maybe you will find that there is a reason that you enjoy some people more than others.

The range of years, 1912 2023, not not cover older people, or if one wishes to go back to earlier times. Just add a series of twelves to the date to bring it within this range, such as for 1896 add 24 and use 1920.

Lines 1912 to 1923 correspond to the first year of each name for ease of checking. Lines 1995, 2005, etc., were also for checking and can be deleted after testing the program.

```
   3 REM public domain program  b
y Dick Wagner, 1991
    4 REM Chinese Fortune Calendar

  1Ø INPUT "Your birth data? Year
s from 1912 to 2Ø23. ";y
  2Ø FOR a=1912 TO 2Ø2Ø STEP 12
  3Ø IF a=y THEN  GO TO 1912
  5Ø NEXT a
 1ØØ FOR b=1913 TO 2Ø21 STEP 12
 11Ø IF b=y THEN  GO TO 1913
 12Ø NEXT b
 2ØØ FOR c=1914 TO 2Ø22 STEP 12
 21Ø IF c=y THEN  GO TO 1914
 22Ø NEXT c
 25Ø FOR d=1915 TO 2Ø23 STEP 12
 255 IF d=y THEN  GO TO 1915
 26Ø NEXT d
 265 FOR e=1916 TO 2Ø12 STEP 12
 27Ø IF e=y THEN  GO TO 1916
 275 NEXT e
 28Ø FOR f=1917 TO 2Ø13 STEP 12
 285 IF f=y THEN  GO TO 1917
 29Ø NEXT f
 295 FOR g=1918 TO 2Ø14 STEP 12
 3ØØ IF g=y THEN  GO TO 1918
 31Ø NEXT g
 315 FOR h=1919 TO 2Ø15 STEP 12
 32Ø IF h=y THEN  GO TO 1919
 325 NEXT h
 33Ø FOR i=192Ø TO 2Ø16 STEP 12
 335 IF i=y THEN  GO TO 192Ø
 34Ø NEXT i
 345 FOR j=1921 TO 2Ø17 STEP 12
 35Ø IF j=y THEN  GO TO 1921
 355 NEXT j
 36Ø FOR k=1922 TO 2Ø18 STEP 12
 365 IF k=y THEN  GO TO 1922
 37Ø NEXT k
 375 FOR l=1923 TO 2Ø19 STEP 12
 38Ø IF l=y THEN  GO TO 1923
 385 NEXT l
191Ø STOP
1912 PRINT "Year of Rat!": GO TO
2ØØØ
1913 PRINT "Year of Ox!": GO TO 2
Ø1Ø
1914 PRINT "Year of Tiger!": GO T
O 2Ø2Ø
1915 PRINT "Year of Hare!": GO TO
 2Ø3Ø
1916 PRINT "Year of Dragon!": GO
TO 2Ø4Ø
1917 PRINT "Year of Snake!": GO T
O 2Ø5Ø
1918 PRINT "Year of Horse!": GO T
O 2Ø6Ø
1919 PRINT "Year of Sheep!": GO T
O 2Ø7Ø
192Ø PRINT "Year of Monkey!": GO
TO 2Ø8Ø
1921 PRINT "Year of Cock!": GO TO
 2Ø9Ø
1922 PRINT "Year of Dog!": GO TO
3ØØØ
1923 PRINT "Year of Boar!": GO TO
 3Ø1Ø
1995 REM Year of Rat
2ØØØ PRINT "Rat: You are noted fo
r charm    and attractivness for
the        opposite sex. You work
hard to  reach goals and get poss
essions.You're thrifty, honest, a
nd wantthings just so. You get an
gry   easily, but manage to look
calm.You get along best with Drag
on, Monkey, Ox."
2ØØ5 NEW : REM Year of Ox
```

```
2010 PRINT "Ox: You are patient,
quiet, and people trust you. Norm
ally you  are easy going, but at
times youmay be stubborn and a li
ttle tooquick to get angry. You h
ave an alert mind and body, and h
ate tofail at anything. You get a
long best with Snake, Cock, Rat."

2015 STOP : REM Year of Tiger
2020 PRINT "Tiger: You are sensit
ive, kind, and a deep thinker. Pe
ople        respect you because you
 are       brave. And you always ge
t creditwhere due. But you can be
 very  short-tempered and sometim
es     have trouble making up your
 mindYou get along best with Hors
e,  Dragon, Dog."
2025 STOP : REM Year of Hare
2030 PRINT "Hare: You are lucky,
talented,  and smart in business.
 You aim  for great things and wi
n them.  You are quiet, kind, and
 seldom lose your temper, but you
 like  to gossip, and you're some
times sad. You get along best wit
h    Sheep, Boar, Dog."
2035 STOP : REM Year of Dragon
2040 PRINT "Dragon: You have good
 health    plus lots of pep and e
nergy. Youget excited easily, and
 may get angry easily too. But pe
ople    trust you because you are
honestbrave, and soft-hearted. Y
ou're nobodys fool though,and you
     never borrow money or make
    speaches. You get along best
   with Rat, Monkey, Cock."
2045 STOP : REM Year of Snake
2050 PRINT "Snake: You are very d
eep, quiet,and wise. You're lucky
 with      money too, and never ha
ve to    worry about it. You are
kind to others, but not apt to be
 very  generous with them. You ar
e a   very determined person and
hate to fail. You get along best
withOx, Cock."
2055 STOP : REM Year of Horse
2060 PRINT "Horse: You are chearf
ul, popularand smart with money,
but you   may be too talkitive an
d showy. You are wise, talented,
and goodwith your hands. Crowds,
fun,   and action attract you, an
d you like the opposite sex. You
get  along best with Tiger, Dog,
    Sheep."
2065 STOP : REM Year of Sheep
2070 PRINT "Sheep: You like nice
things, andmay become an artist o
r musicianYou have strong beliefs
. Yet youmay be shy, timid, and s
omewhat puzzled by life. Your abi
lities make money for you. You ge
t     along best with Hare, Boar
and  Horse."
2075 STOP : REM Year of Monkey
2080 PRINT "Monkey: You're apt to
 be clever,and may have a bit of
genius.   You like to invent thin
gs, solvehard problems and think
up new  ideas. You're a good thin
ker andwant to know all about thi
ngs.  You may become famous. You
get  along best with Rat, Dragon.
"
2085 STOP : REM Year of Cock
2090 PRINT "Cock: You're a deep t
hinker withmuch ability and talen
t.  You   like to keep busy, you
try hard and you hate to fail. Yo
u'd      rather work hard by yours
elf     than with others and your
      fortunes swing high, or low
. Youget along best with Ox, Snak
e,  Dagon."
2095 STOP : REM Year of Dog
3000 PRINT "Dog: You're truthful,
 loyal, andpeople trust you, beca
use you   stand up for what is ri
ght, and you can keep secrets. Yo
u don't care much for wealth, but
 you   seem to have enough. You g
et     along best with Horse, Tige
r,   Hare."
3005 STOP : REM Year of Boar
3010 PRINT "Boar: You're quiet an
d study a  lot because you want k
nowledge. You do all things with
all your strength. You don't make
 many   friends, but you keep tho
se you make because you're honest
, kindand true to others. You get
     along best with Hare, Sheep.
"
9990 STOP
9999 RANDOMIZE USR 100: SAVE "for
tun.B1" LINE 1
```

ILLUSIONS

by Tad Hendrickson
(Mods by Rod Gowen)

I want to apologize to any of you who went to the trouble to type in this program out of the last issue of The Plotter and found that it would not run! I guarantee that this time it will run for you. Please try it again and see.

I have played with it a little and have put together a couple of alterations that you may enjoy trying.

After typing in the listing and running it, BREAK to BASIC and try changing the display in one of the following ways:

Try adding ":CLS" to the end of Line 280, PRETTY FLASHY, HUH!

Try deleting the ":CLS" FROM lINE 70. watch this one if you can!!

Be sure to try some of the direction changes as described in the documentation in last months' Plotter.

I think that you will enjoy this one. Try the changes and try some of your own. Have fun!

```
    5 REM ILLUSION
         By Tad Hendrickson
        Mods by Rod Gowen
   10 PAPER 0: BORDER 0: RESTORE
   20 FOR a=65510  TO 65521: READ
 byte: POKE a,byte: NEXT a
   30 FOR a=65522  TO 65533: READ
 byte: POKE a,byte: NEXT a
   40 LET p2=PI/2: LET sp=p2/9
   50 LET ep=sp/5: LET dt=PI/15
   60 LET b=1
   70 FOR v=1 TO 5
   80 INK 7: CLS
   90 PRINT AT 21,5;"...constructi
 ng view #";v
  100 LET dp=ep*v
  110 FOR i=dp TO PI STEP sp
  120 LET a=COS (i)
  130 LET first=1
  140 FOR t=0 TO PI STEP dt
  150 LET x=a*SIN (t)
  160 LET y=b*COS (t)
  170 LET px=x*60+127
  180 LET py=y*60+115
  190 IF first  THEN  PLOT px,py:
 LET first=0
  200 IF NOT first THEN  DRAW (px-
 (PEEK 23677)),(py-(PEEK 23678))
  210 NEXT t
  220 NEXT i
  230 PRINT AT 21,0;"
                 "
  240 READ code
  250 POKE 65512,code
  260 RANDOMIZE USR 65510
  270 NEXT v
  280 CLS
  290 RESTORE 530
  300 FOR v=1 TO 5: REM  CLS
  310 READ code
  320 POKE 65527,code
  330 RANDOMIZE USR 65522
  340 NEXT v
  350 GO TO 290
  510 DATA 17,208,0,33,0,64,1,0,16
 ,237,176,201
  520 DATA 17,0,64,33,208,0,1,0,16
 ,237,176,201
  530 DATA 132,148,164,180,196
  600 REM
  610 RANDOMIZE USR 100: GO TO 0
  620 SAVE "illusn" LINE 10
```

CIRCLES

This cone design for the 2068 makes good use of CIRCLE to generate a design quickly. Users of the 2068 will recall that in line 120 the first number gives the circle center location x axis, the second number is the center y axis, and the third is the circle radius. The trick is to move the center ypward at a rate less than the circles increase in diameter. Try varying the last number in small increments.

PROGRAM: CIRCLES

```
100 FOR n=1 TO 50 STEP 3
120 CIRCLE 100, 20+2*n,2+n
130 NEXT n
```

```
  10 GO SUB 1000: REM By James
 Edwards from The Plotter/Alts
by Jack Armstrong 3-86
  20 FOR i=1 TO 8: PRINT AT i,22;
 PAPER 7;"        ": NEXT i
  30 FOR i=0 TO 7: PRINT AT i,0;"
AAAAAAAA      ": NEXT i: REM UseGr
ahpic A
  40 LET r=0: LET o=0: LET a=1
  50 FOR i=1 TO 8: PRINT AT i,21;
I: NEXT i
  60 PRINT AT 0,22;"ABCDEFGH";AT
9,22;"ABCDEFGH"
  70 FOR I=1 TO 8: PRINT AT I,30;
I: NEXT I
  80 PRINT AT 9,0;"PRESS 9 TO FIL
L";AT 10,0;"PRESS 0 TO ERASE";AT
11,0;"PRESS F WHEN FINISHED";AT 1
2,0;"Use Arrow Keys to Move Curso
r": PRINT
  90 PRINT "[A]...Alter Character
"
 100 PRINT "[E]...Erase Character
"
 110 PRINT "[N]...For New Picture
"
 120 PRINT "[P]...Copy Screen"
 130 PRINT ' INK 2; PAPER 7;"NOTE
: Copy Screen or Write Down the D
ATA Info After Each New UDG"
 140 PRINT  OVER 1;AT r,o;"*"
 150 IF INKEY$="" THEN  GO TO 150

 160 LET m$=INKEY$: BEEP .01,20
 170 IF m$="f" OR m$="F" THEN  GO
 TO 260
 180 IF m$="9" THEN  PRINT AT r,o
;"@": BEEP .05,-10: GO TO 140
 190 IF m$="0" THEN  PRINT AT r,o
;"A": BEEP .05,50: GO TO 140: REM
 Use Graphic A
 200 LET pr=r: LET po=o
 210 LET r=r+(m$="6")-(m$="7")
 220 LET o=o+(m$="8")-(m$="5")
 230 IF o<0 OR o>7 THEN  LET o=po

 240 IF r<0 OR r>7 THEN  LET r=pr

 250 PAUSE 6: PRINT  OVER 1;AT pr
,po;"*": GO TO 140
 260 LET byte=0: DIM d(8)
 270 PRINT  OVER 1;AT r,o;"*"
 280 FOR a=171 TO 115 STEP -8
 290 LET bit=8
 300 FOR i=4 TO 60 STEP 8
 310 LET bit=bit-1
 320 IF POINT (i,a) THEN  LET d(b
yte+1)=d(byte+1)+2^bit
 330 PRINT AT byte,10;"   ";AT byt
e,10;d(byte+1): NEXT i
 340 POKE 65376+byte,d(byte+1)
 350 LET byte=byte+1
 360 NEXT a
 370 LET a=22: LET b=23: LET c=24
: LET d=25: LET e=26: LET f=27: L
ET g=28: LET h=29
 380 POKE 23658,8
 390 INPUT  PAPER 7;"Paper Color:
 ";cp
 400 INPUT  PAPER 7;"Color of Ink
: ";q
 410 PRINT AT 21,0; INK 2; FLASH
1;"Print At?"
 420 INPUT  PAPER 7;"Row: ";y; PA
PER 7;"Column: ";x
 430 PRINT AT y,x; PAPER cp; INK
q;"B": REM Use Graphic B
 440 PRINT AT 21,0; INK 6;"
   "
 450 POKE 23658,0
 460 FOR i=13 TO 17: PRINT AT i,2
3; PAPER 6; INK 6; FLASH 0;"
   ": NEXT i
 470 IF INKEY$="" THEN  GO TO 460

 480 IF INKEY$="a" THEN  GO TO 14
0
 490 IF INKEY$="e" THEN  GO TO 30

 500 IF INKEY$="p" THEN  GO TO 53
0
 510 IF INKEY$="n" THEN  RUN
 520 GO TO 470
 530 PRINT AT 13,23;"After";AT 14
,23;"BREAK";AT 15,23;"Enter";AT 1
6,23;"Direct";AT 17,23; PAPER 2;
INK 7; FLASH 1;"GOTO 460"
 540 PRINT AT 21,0;"Press ENTER t
o COPY": PAUSE 0
 550 PRINT AT 21,0;"
      ": COPY
 560 GO TO 460
 990 STOP
1000 REM
1010 BORDER 5: PAPER 6: INK 0: CL
S : RESTORE : REM U may want tous
e BRIGHT also
1020 FOR k=0 TO 7: READ d
1030 POKE 65368+k,d: NEXT k
1040 DATA 255,129,129,129,129,129
,129,255
1050 REM
```

```
1060 REM 65368 is the address of
     the start of the UDG area
1090 RETURN
1100 REM
1110 CLS : PRINT '''''TAB 8;"Now
SAVEing ""udg"""
1120 RANDOMIZE USR 100: GO TO 0:
REM This directs the SAVE to    d
rive 0
1150 SAVE "udg" LINE 10
1160  REM This program's altered
      from the original printing
     in the Plotter April 1986
1170  REM  U can substitute any
      other method of SAVE for
     your own equipment
```

```
  10  LET a$="ABC"
  20 IF A$>"GO" THEN  PRINT "ABC
IS GREATER THAN GO"
  30 IF "GO+0">A$ THEN  PRINT "GO
0 IS GREATER THAN ABC"
  40 IF "SALLY"<"GLENDA" THEN  GO
 TO 1000
  50 IF "GLENDA">"SALLY" THEN  GO
 TO 1010
  60 IF "SALLY">"GLENDA" THEN  PR
INT "SALLY IS GREATER"
  80 STOP
1000 PRINT "SALLY = 389 WHILE GLE
NDA = 427"
1005 REM FOR T/S 1000,    SALLY=2
44 AND GLENDA=265
1007 STOP
1010 PRINT "GLENDA IS GREATER THA
N SALLY"
1050
1060 STOP
1070 REM LINES 60 AND 80 WERE
    ADDED TO THE ORIGINAL PRO-
   GRAM AS PRINTED BECAUSE IT
  DID NOT GIVE THE PROPER
 RESULT. THE PROOF OF THE
EVALUATION LIES IN THE TRUE     C
OMPARISION OF SALLY AND        GL
ENDA...THE CODE OF THE        FIR
ST CHARACTER IS THE TEST     OF V
ALUE; IF THEY ARE THE       SAME,
 THEN THE NEXT CHAR-        ACTER
IS COMPARED.
1080 REM THE REST OF THE PROGRAM
    IS IN DOUBT SINCE THE VALUE
   OF S IS 83 & OF G IS 71. AN
  EVALUATION IS TAKEN CHARAC-
 TER BY CHARACTER, NOT AS A
WHOLE.
1090 REM THEREFORE Z>A, B>A, F<H
    DD>DA, ETC.
1100 SAVE "compar"
```

```
  10  REM SCREEN FORMAT ROUTINE
  20 REM FROM JACK ARMSTRONG
  25 REM This works on the 2068
  30 REM Note the positioning is
    tied to the loop #'s
  35 BORDER 5: PAPER 6: INK 1: CL
S
  40 LET x$="CLACKAMAS"
  50 PRINT : PRINT : PRINT : PRIN
T
  60 REM This loop prints right
  70 FOR k=1 TO LEN x$: PRINT AT
4,11+k;x$(k);: NEXT k
  80 REM This loop prints down
  90 FOR k=LEN x$ TO 1 STEP -1
 100 PRINT AT k-13,20;x$(k)
 110 NEXT k
 120 REM This loop prints back
    to the left
 130 FOR k=1 TO LEN x$
 140 PRINT AT 12,21-k;x$(k)
 150 NEXT k
 160 REM This loop prints up
 170 FOR k=LEN x$ TO 1 STEP -1
 180 PRINT AT k+3,12;x$(k)
 190 NEXT k
 200 PRINT AT 4,12;"C";AT 5,13;"C
ounty";AT 6,14;"Area";AT 7,15;"T"
;AT 8,16;"S";AT 9,14;"Users";AT 1
0,14;"Group"
 210 PLOT 87,151: DRAW 88,0: DRAW
 0,-88: DRAW -88,0: DRAW 0,88
 220 PLOT 84,153: DRAW 0,-92: DRA
W 94,0: DRAW 0,92: DRAW -94,0
 230 PRINT AT 3,11;"5";AT 3,21;"8
";AT 13,21;">";AT 13,11;"?"
 240 PLOT 123,107: DRAW 9,9
 250 REM This could be adapted
    to the 1000 except for the
   PLOTs and DRAWs
 260 STOP
 300 SAVE "loopformat" LINE 10
 310 REM the next line would be
    appropriate for the Larken
   system
 400 REM  RANDOMIZE USR 100: SAVE
 "lp4mat.B1" LINE 10
```

```
   1 REM There is something
    missing or wrong about this
   program. Not only does it
  not work properly, it does
 the same thing every time,
there is nothing random        a
bout it and it does not       pr
int a full square
   2 REM There must be something
    missing to make it work
  10 REM THERE MUST BE AN ODD
    NUMBER OF COLUMNS/ROWS
 100 DIM M(9,9)
 110 PRINT AT 10,0;"Number of Col
umns/Rows?"
 115 INPUT n
 120 CLS
 130 IF n/2=INT (n/2) THEN  GO TO
 110
 140 PRINT AT 0,0;"  MAGIC NUMBER
S";AT 1,0;"   ROWS/COLUMNS."
 150 REM
 160 LET C1=0
 170 LET C=INT (N/2)+1
 180 LET R=1
 185 LET C1=C1+1
 190 PRINT AT R*2+1,C*3;C1
 200 IF C1=INT (N^2) THEN  GO TO
330
 210 IF C1/N<>INT (C1/N) THEN  GO
 TO 240
 220 LET R=R+1
 230 GO TO 185
 240 LET C=C+1
 250 IF C<=N THEN  GO TO 290
 260 LET C=1
 270 LET R=R-1
 280 GO TO 185
 290 LET R=R-1
 300 REM  COPY
 310 LET R=N
 320 GO TO 185
 330 LET T=0
 340 FOR I=1 TO N
 350 LET T=T+M(I,1)
 360 NEXT I
 370 PRINT AT 0,0;T
 375 STOP
 380 COPY
 390 PAUSE 250
 400 LLIST
 410 REM  NEW
 415 REM CHANGE THE SAVE LINE TO
    SUIT YOUR EQUIPMENT
 420 STOP
 430 SAVE "magic" LINE 10
```

```
   1 REM OHMS BIG LAW
   2 REM PROGRAM BY DUANE HEWITT
   3 REM 1986, AND IN 1992, ALTS
 BY JACK ARMSTRONG FOR
   4 REM INCLUSION IN "BEST OF"
   5 BORDER 5: INK 1: PAPER 6: C
LS
   7 GO SUB 6000
   8 CLS
  10 PRINT AT 1,5;"ENTER FORMULA
 YOU NEED"
  30 PRINT AT 4,12; PAPER 0; INK
 7;"  MENU  ";AT 5,12;"         "
  40 PRINT AT 6,12; PAPER 0; INK
 7;"1- E=I*R";AT 7,12;"         "
  50 PRINT AT 8,12; PAPER 0; INK
 7;"2- I=E/R";AT 9,12;"         "
  60 PRINT AT 10,12; PAPER 0; IN
K 7;"3- R=E/I"
  65 PRINT AT 15,0;"R=RESISTANCE
 E=VOLTAGE I=CURRENT"
  66 PRINT AT 16,0;"    ""OHMS""
  ""VOLTS""   ""AMPS"""
  70 INPUT A
  80 IF A>3 THEN GO TO 70
  90 IF A=1 THEN GO TO 1000
 100 IF A=2 THEN GO TO 2000
 110 IF A=3 THEN GO TO 3000
1000 REM
1001 CLS
1005 PRINT AT 1,13;"E=I*R"
1010 PRINT AT 2,7; PAPER 5; INK
2;" ENTER VALUE OF I "
1015 PRINT AT 15,0;"R=RESISTANCE
 E=VOLTAGE I=CURRENT"
1016 PRINT AT 16,0;"    ""OHMS""
  ""VOLTS""   ""AMPS"""
1020 INPUT I: PRINT AT 4,0;"CURR
ENT  =   ";I;AT 4,21;"AMPS"
1030 PRINT AT 2,2; PAPER 5; INK
2;" ENTER VALUE OF RESISTANCE "
1040 INPUT R: PRINT AT 7,0;"RESI
STANCE=    ";R;AT 7,21;"OHMS"
1045 PRINT AT 2,0;"
                     "

1050 LET E=I*R
1060 PRINT AT 10,0;"R=RESISTANCE
 E=VOLTAGE I=CURRENT"
1070 PRINT AT 15,0;"R=RESISTANCE
 E=VOLTAGE I=CURRENT"
1071 PRINT AT 16,0;"    ""OHMS""
  ""VOLTS""   ""AMPS"""
1075 PRINT AT 20,0;"    PRESS AN
Y KEY TO CONTINUE    "
1080 PAUSE 4E4
1105 GO TO 4010
2000 REM
```

```
2001 CLS
2005 PRINT AT 1,13;"I=E/R"
2006 PRINT AT 15,0;"R=RESISTANCE
E=VOLTAGE I=CURRENT"
2007 PRINT AT 16,0;"    ""OHMS""
    ""VOLTS""   ""AMPS"""
2010 PRINT AT 2,7; PAPER 5; INK 2
;" ENTER VALUE OF E "
2020 INPUT E: PRINT AT 4,0;"VOLTA
GE   =   ";E;"    VOLTS"
2030 PRINT AT 2,2; PAPER 5; INK 2
;" ENTER VALUE OF RESISTANCE "
2040 INPUT R: PRINT AT 7,0;"RESIS
TANCE=   ";R;AT 7,21;"OHMS"
2045 PRINT AT 2,0;"
                   "

2050 LET I=E/R
2060 PRINT AT 10,0;"CURRENT   =
 ";I;AT 10,21;"AMPS"
2070 PRINT AT 20,0;"   PRESS ANY
KEY TO CONTINUE    "
2100 PAUSE 4E4
2105 GO TO 4010
3000 REM
3001 CLS
3005 PRINT AT 1,13;"R=E/I"
3006 PRINT AT 15,0;"R=RESISTANCE
E=VOLTAGE I=CURRENT"
3007 PRINT AT 16,0;"    ""OHMS""
    ""VOLTS""   ""AMPS"""
3010 PRINT AT 2,7; PAPER 5; INK 2
;" ENTER VALUE OF I "
3020 INPUT I: PRINT AT 4,0;"CURRE
NT   =   ";I;AT 4,21;"AMPS"
3030 PRINT AT 2,7; PAPER 5; INK 2
;" ENTER VALUE OF E "
3040 INPUT E: PRINT AT 7,0;"VOLTA
GE   =   ";E;AT 7,21;"VOLTS"
3045 PRINT AT 2,0;"
                   "

3050 LET R=E/I
3060 PRINT AT 10,0;"RESISTANCE=
 ";R;AT 10,21;"OHMS"
3100 PAUSE 4E4
3105 GO TO 4010
4000 REM
4010 PRINT AT 15,0;"  DO YOU WANT
 ANOTHER FORMULA?  "
4011 PRINT AT 16,0;"
                   "

4012 PRINT AT 20,0;"
                   "

4015 PRINT AT 17,14;"Y/N"
4020 INPUT W$
4030 IF W$="Y" OR W$="y" THEN  CL
```

```
4500 PRINT AT 17,5;"IT'S NICE TO
BE NEEDED "
4501 PRINT AT 19,11;"THANK YOU!"
4505 PAUSE 4E4
4506 STOP
6000  PRINT AT 1,0;"THIS PROGRAM
IS VERY HANDY WHEN YOU WANT TO FI
GURE A LITTLE OHMSLAW AND DO NOT
KNOW ALL THERE ISTO KNOW ABOUT OH
MS LAW."
6005 PRINT AT 6,0;"FOLLOW THE MEN
U"
6010 PRINT AT 9,0;"CHOOSE THE FOR
MULA FOR ""E"" IF   THE UNKNOWN I
S VOLTAGE"
6020 PRINT AT 13,0;"CHOOSE THE FO
RMULA FOR ""I"" IF   THE UNKNOWN
IS CURRENT"
6030 PRINT AT 17,0;"CHOOSE THE FO
RMULA FOR ""R"" IF   THE UNKNOWN
IS RESISTANCE"
6035 PRINT AT 20,0;"PUSH ANY KEY
TO CONTINUE"
6040 PAUSE 4E4
6050 RETURN
6100 STOP
7000 SAVE "ohmlaw" LINE 1
7010 BEEP .5,8: GO TO 1
```

```
  10 REM
  20 REM DEMO OF HIDDEN CODES
  30 PRINT "This is FLASH"
  40 PRINT "THIS IS PAPER & INK"
  50 PRINT "THIS IS INVERSE"
  60 PRINT "THIS IS RED"
  70 PRINT "THIS IS BLUE"
  80 PRINT "THIS IS MAGENTA"
  90 PRINT "THIS IS GREEN"
 100 PRINT "THIS IS CYAN"
 110 PRINT "    THIS IS YELLOW"
 120 PRINT "THI S IS INK"
 130 PRINT "THIS IS INVERSE"
 140 PRINT "THIS IS PAPER"
 150 PRINT "THIS IS PAPER"
 160 PRINT "THIS IS PAPER"
 170 PRINT "THIS IS PAPER"
 180 PRINT "THIS IS PAPER"
 190 PRINT "    THIS IS PAPER"
 200 PRINT "THIS IS PAPER"
 210 PRINT "THI S IS INK and FLAS
H"
 250 STOP
1000 REM
1010 SAVE "hidden"
```

(Editor's Note:-
See article Page 29)

SEE PAGE 73 FOR INSTRUCTIONS

```
   1 REM   The Checkwriter
   2 REM 1987 by Syd Wyncoop
   3 BORDER 5: PAPER 6: INK 1: CL
S
   4 LET getdigit=500
  10 REM Set-up check writing
              strings
  11 REM Set-up One$
  12 DIM O$(10,5)
  13 LET O$(1)="VOID*"
  14 LET O$(2)="One "+CHR$ 8
  15 LET O$(3)="Two "+CHR$ 8
  16 LET O$(4)="Three"
  17 LET O$(5)="Four"
  18 LET O$(6)="Five"
  19 LET O$(7)="Six "+CHR$ 8
  20 LET O$(8)="Seven"
  21 LET O$(9)="Eight"
  22 LET O$(10)="9"
  23 REM Set-up Ten$
  24 DIM T$(10,8)
  25 LET T$(1)="**VOID**"
  26 LET T$(2)="**VOID**"
  27 LET T$(3)=" Twenty "+CHR$ 8
  28 LET T$(4)=" Thirty"+CHR$ 8
  29 LET T$(5)=" Forty "+CHR$ 8
  30 LET T$(6)=" Fifty "+CHR$ 8
  31 LET T$(7)=" Sixty "+CHR$ 8
  32 LET T$(8)=" Seventy "
  33 LET T$(9)=" Eighty "
  34 LET T$(10)=" Ninety "
  35 REM Set-up Teen$
  36 DIM N$(10,10)
  37 LET N$(1)=" Ten  "+CHR$ 8+CH
R$ 8
  38 LET N$(2)=" Eleven "+CHR$ 8
  39 LET N$(3)=" Twelve "+CHR$ 8
  40 LET N$(4)=" Thirteen"
  41 LET N$(5)=" Fourteen"
  42 LET N$(6)=" Fifteen "+CHR$ 8

  43 LET N$(7)=" Sixteen "+CHR$ 8

  44 LET N$(8)=" Seventeen"
  45 LET N$(9)=" Eighteen"
  46 LET N$(10)=" Nineteen"
  99
 100 REM Test Program
 101 INPUT "Enter the number to w
rite"' LINE C$
 102 GO SUB 110
 103 PRINT K$
 104 GO TO 100
 110 REM Enter the subroutine
        with C$=number to write
 111 REM Pad-out number to two
        decimal places and make
        sure no leading spaces
        or leading zeroes
 112 LET A$=C$+".00"
 113 IF A$(1)=" " THEN  LET A$=A$
(2 TO ): GO TO 113
 115 LET C=1
 120 FOR I=1 TO LEN a$
 121 IF A$(I)="." THEN  GO TO 124

 122 LET C=C+1
 123 NEXT I
 124 LET C=C+2
 125 LET A$=A$( TO C)
 126 IF A$(1)="0" THEN  LET A$=A$
(2 TO ): GO TO 126
 127 LET C=LEN A$: IF A$(C)="." T
HEN  LET A$(C)="0"
 130 REM Convert to String
 131 LET C=LEN A$
 132 LET K$="***"
 133 LET FLAG=0
 134 IF C=3 THEN  GO TO 220
 135 IF C<10 THEN  GO TO 140
 136 GO SUB GETDIGIT: LET K$=O$(A
)+" Million, "
 140 IF C<9 THEN  GO TO 150
 141 GO SUB GETDIGIT: IF A=1 THEN
  GO TO 150
 142 LET K$=K$+O$(A)+" Hundred":
LET FLAG=1
 150 IF C<8 THEN  GO TO 160
 151 GO SUB GETDIGIT: IF A=1 THEN
  LET K$=K$+" ": GO TO 160
 152 IF A=2 THEN  GO SUB GETDIGIT
: LET K$=K$+N$(A): LET FLAG=1: GO
 TO 170
 153 LET K$=K$+T$(A): LET FLAG=1
 154 GO SUB GETDIGIT: IF A=1 THEN
  GO TO 170
 155 LET K$=K$+"-"+O$(A): LET FLA
G=1: GO TO 170
 160 IF C<7 THEN  GO TO 170
 161 GO SUB GETDIGIT: IF A=1 THEN
  GO TO 170
 162 LET K$=K$+O$(A): LET FLAG=1
 170 IF FLAG=1 THEN  LET K$=K$+"
Thousand, ": LET FLAG=0
 171 IF C<6 THEN  GO TO 180
 172 GO SUB GETDIGIT: IF A=1 THEN
  GO TO 180
 173 LET K$=K$+O$(A)+" Hundred":
LET FLAG=1
 180 IF C<5 THEN  GO TO 190
 181 GO SUB GETDIGIT: IF A=1 AND
FLAG=1 THEN  LET K$=K$+" ": LET F
LAG=0
 182 IF A=1 THEN  GO TO 190
 183 IF A=2 THEN  GO SUB GETDIGIT
: LET K$=K$+N$(A): GO TO 210
 184 LET K$=K$+T$(A)
 185 GO SUB GETDIGIT: IF A=1 THEN
  GO TO 210
 186 LET K$=K$+"-"+O$(A): GO TO 2
10
 190 IF C<4 THEN  GO TO 210
 191 GO SUB GETDIGIT: IF A=1 THEN
  GO TO 210
 192 LET K$=K$+O$(A)
 210 IF C<=3 THEN  GO TO 220
 211 LET K$=K$+" AND "
 220 LET A$=A$(LEN A$-1 TO )
 221 IF A$="00" THEN  LET K$=K$+"
 No": GO TO 240
 223 GO SUB GETDIGIT: LET K$=K$+S
TR$ (A-1)
 230 LET K$=K$+A$
 240 LET K$=K$+"/100***"
 241 REM Most Checks have the
    word "Dollars" printed on
   them but, you could add it
  here if needed as follows:
 242 REM  LET k$=k$+" Dollars"
 250 RETURN
 500 REM GETDIGIT
 501 LET A=VAL A$(1)+1
 502 LET A$=A$(2 TO )
 503 RETURN
9989 STOP
9990 SAVE "checks" LINE 1
```

```
   1 REM Bold - from Bob Evans
   2 REM Unknown source
  10 STOP
9000 RESTORE 9040
9005 FOR b=1 TO 29
9010 READ a
9020 POKE 65267+b,a
9030 NEXT b
9040 DATA 17,0,221,213,1,0,3,42,5
4,92,36,126,167,31,182,18,35,19,1
3,32,246,16,244,225,37,34,54,92,2
01
9050 GO TO 10
9500 CLS
9520 IF PEEK 23607=60 THEN  RANDO
MIZE USR 65268: GO TO 10
9530 POKE 23607,60
9540 GO TO 10
9550 STOP
9560 SAVE "bold"
9565 REM Change the SAVE to suit
    your equipment, this=Larken
9570 STOP
```

```
  10 REM 1000 PROG. LOAN STATUS
  20 REM CONVERTED TO 2068, BUT
    OK FOR 1000-ADD SLOW WHERE
   NOTED
  30 REM USE ** ON THE 1000 FOR
    TO THE POWER OF: ^
  40 REM THIS PROGRAM DOES NOT
    TAKE INTO ACCOUNT IMPOUNDS
   FOR TAXES OR INSURANCE IN-
  CLUDED IN PAYMENTS OR FOR
 IMPOUNDS OF ANY OTHER SORT
  50 REM YOU'D HAVE TO FIGURE
    THOSE SEPARATELY
  60 BORDER 5: PAPER 6: INK 1: CL
S
 100 PRINT TAB 11;"LOAN STATUS"
 110 PRINT
 120 PRINT ,,,,"  THIS PROGRAM WI
LL COMPUTE THE"
 130 PRINT " APPROXIMATE INTEREST
 PAID ON A"
 135 PRINT " LOAN FOR ANY GIVEN P
ERIOD, AND"
 140 PRINT " SUPPLY THE APPROXIMA
TE PRINCI-"
 145 PRINT " PAL BALANCE REMAININ
G."
 150 PRINT ,,,,"    PRESS ANY KEY
 TO START..."
 155 PAUSE 4E4
 160 CLS
 165 PRINT "  WHAT WAS THE ORIGIN
AL TOTAL     NUMBER OF PAYMENTS?"

 170 INPUT N
 175 PRINT AT 1,27;N
 180 PRINT ,," WHAT IS THE PAYMEN
T NUMBER OF   THE FIRST PAYMENT I
N THE         SUBJECT PERIOD?"
 185 INPUT N1
 190 PRINT AT 5,27;N1
 195 LET N1=N1-1
 200 PRINT ,,"WHAT IS THE PAYMENT
 NUMBER OF   THE LAST PAYMENT IN
THE SUBJECT PERIOD?"
 205 INPUT N2
 210 PRINT AT 9,27;N2
 215 PRINT ,,"PLEASE ENTER THE NO
RMAL MONTHLY PAYMENT AMOUNT."
 220 INPUT M
 225 PRINT AT 12,27;M
 230 PRINT ,,"PLEASE ENTER THE AN
NUAL PERCENT-AGE RATE."
 235 INPUT R1
 240 PRINT AT 15,27;R1
 245 PAUSE 120: REM CHANGE THE
```

```
   PAUSE TO 400 FOR THE 1000
 250 CLS
 255 LET R=R1/1200
 260 PRINT ,,,,
 265 REM SLOW GOES HERE
 270 LET I=M*(N2-N1-((((1+R)^(N2-N
))/R)+(((1+R)^(N1-N))/R))
 290 PRINT "THE TOTAL INTEREST PA
ID DURING  THE PERIOD IS $";I
 295 LET V=(M/R)*(1-(1+R)^(N2-N))

 300 PRINT ,,,,"THE UNPAID PRINCI
PAL BALANCE    AFTER PAYMENT NUMB
ER ";N2
 305 PRINT "IS $";V
 310 PRINT ,,,,,,"WOULD YOU LIKE
TO SOLVE ANOTHER PROBLEM?"
 315 POKE 23658,8: REM THIS FOR
    THE 2068; FORCES CAPITALS
 320 INPUT Z$
 325 IF Z$(1)="Y" THEN  GO TO 160

 330 CLS
 335 PRINT ,,,,"THANK YOU. I HOPE
 I HAVE BEEN OFSOME HELP TO YOU."

 340 STOP
1000 REM
1010 SAVE "lonstus"
1020 GO TO 10
```

```
   1 REM CR w/LF..POKE 64460,10
   2 REM CR w/o LF..POKE 64460,0
   3 REM Width......POKE 64459,Wi
dth
   4 REM LPRINT mode...POKE 64456
,1
   5 REM Cntrl Mode....POKE 64456
,0
   6 REM Turn on..POKE 26703,205
       & POKE 26704,251
   7 REM Turn off..POKE 26703,0
       & POKE 26704,5
  10 REM  CLEAR 64455: LOAD ""COD
E
  11 REM Line 10 is to LOAD yo
    printer driver CODE - you
   must have already saved
  your own customized version
 Syd mentions AERCO, but you
need to use what is right       f
or your own equipment.
  20 BORDER 0: PAPER 0: INK 9: CL
S : REM set screen attributes
  30 POKE 23658,8: INPUT AT 0,0;"
Please select:"'"(P)rinter or (T)
S 2040:"; LINE a$: REM select a p
rinter
  40 IF a$="T" THEN  POKE 26703,0
: POKE 26704,5: GO TO 130
  41 REM Line 40 gives correct
     channel info for 2040
  50 IF a$="P" THEN  POKE 26703,2
05: POKE 26704,251: GO TO 70
  60 GO TO 30: REM trap entry
     errors, note the POKE in
    line 30 to ensure CAPS
  70 INPUT AT 0,0;"Length of line
 to be printed?"' LINE a$
  71 REM Get width of printer
     line
  80 IF VAL a$<1 OR VAL a$>150 TH
EN  GO TO 70: REM error trap
  90 POKE 64459,VAL a$-1: REM set
     line length to print
 100 INPUT "Do you need a line fe
ed after   the carriage returns?
(Y/N"; LINE a$
 101 REM This depends on how the
     micro-switches are set on
    your printer. Answer Yes if
  not sure.
 110 IF a$="N" THEN  POKE 64460,0
: GO TO 300
 111 REM This gives: no LF after
     CR
 120 POKE 64460,10: REM gives LF
     after CR
 130 PRINT "OK, we should be set
up correct for the printer we sel
ected."'''"You may re-select print
er optionby re-running the progra
m."
 140 PRINT : PRINT "You should no
w study the programlisting from l
ine 300 on to de- termine how to
use the printer- driver to print
text from BASIC.": LIST 300
 150 REM
 300 REM The following program
     lines demonstrate method
    used to print to a large
  printer from BASIC with the
 AERCO printer driver (it is
supplied with interface)
 310 LET mode=64456: LET print=1:
 LET cntrl=0
 311 REM variable names to keep
     everything clear & simple
 320 LPRINT "This is a test line.
 It is printed exactly as typed."
 330 POKE mode,cntrl: LPRINT CHR$
 14;
 331 REM This sends control code
    for large letters to prntr.
 340 POKE mode,print: LPRINT "We
are printing large letters."
 341 REM We are back in text
    mode, but are printing
   large letters.
 350 POKE mode,cntrl: LPRINT CHR$
 15;: REM This sends code for
   small letters to printer
 360 POKE mode,print: LPRINT "We
should now be printing in condens
ed mode."
 361 REM We are back in text
    mode, but are printing
   condensed letters.
 370 POKE mode,cntrl: LPRINT CHR$
 18;
 371 REM We have to turn off the
    shift-in mode but shift-out
   is only active for one line
 380 POKE mode,print: LPRINT "Thi
s is a test line. It is printed e
xactly as typed.":
 381 REM We are back in normal
    mode
 390 POKE mode,cntrl: LPRINT CHR$
 27;CHR$ 52: REM This turns on
   Italics
 400 POKE mode,print: LPRINT "Thi
s is Italic mode, your printer ca
n do it": REM Print in Italics
 410 POKE mode,cntrl: LPRINT CHR$
 27;CHR$ 53: REM Turn off
   Italics
 420 LPRINT "...and that is the e
nd of our examples."
 450 STOP
 500 REM
 510 SAVE "80col"
 511 REM Modify the SAVE routine
         for your system
```

```
   5 REM This is an updated ver-s
ion of Dick Wagner's PRINTUSINGro
utine showing a demo of how itwil
l change your input and printit o
ut in an orderly fashion fora nea
t appearance. It shows how ordina
ry input prints, then willprint t
he same material in for- matted f
ashion. Revisions are byJack Arms
trong on 8-87.
   6 REM Additional correction
     done while keyboarding on
   5/17/92 to get the total to
  print a trailing zero when
  the total is xxxx.x0
   7 REM see the subroutine at
     4000 and the use of t$
  10 INK 0: PAPER 6: BORDER 4: CL
S
  20 PRINT '"This program will in
take your   numbers and print the
m out in   a ""PRINTUSING"" Forma
t even if   they have no leading
digit and  even if they have no d
ecimal;   it will handle numbers
up to 9  characters long (includi
ng the  decimal point.)"
  30 PRINT  INK 1;'" PRESS A KEY
TO START...": PAUSE 0: CLS
  40 PRINT '"How many numbers wil
l we use?": INPUT many: PRINT ' I
NK 2;many;" numbers"
  50 PRINT '"How long is the long
est number? (long=total character
s includingthe decimal point if i
t has one)": INPUT long: PRINT '
INK 2;"Longest number:";long
  60 POKE 23658,8: INPUT "Clear t
he screen? Y/N "; LINE z$: IF z$=
"Y" THEN  CLS
  70 LET tot=0: DIM m(many): LET
i$="######.##"
  80 FOR l=1 TO many
  90 INPUT ("INPUT #";l;" of ";ma
ny);" ";m(l)
 100 LET tot=tot+m(l): NEXT l
 110 PRINT ''
 120 FOR l=1 TO many
 130 PRINT  INK 2;;m(l);TAB 3+lon
g;("Old Format" AND l=1)
 140 NEXT l
 145 PRINT "---------": PRINT  IN
K 2;tot
 150 PRINT
 160 FOR l=1 TO many
 170 LET xi=m(l)
 180 GO SUB 2000
 190 PRINT  INK 1;x$;TAB 3+long;(
"PrintUsing Format" AND l=1)
 200 NEXT l
 205 LET t$=STR$ tot: GO SUB 4000

 210 PRINT "---------": PRINT  IN
K 1;TAB (9-LEN T$);T$
 300 STOP
2000 REM
```

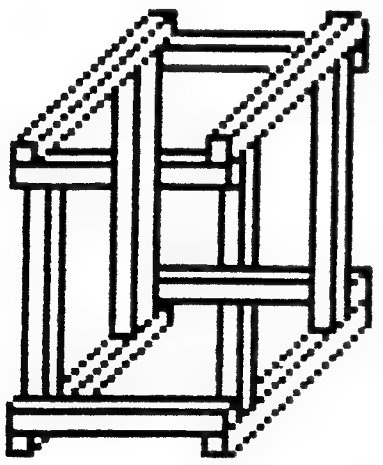
```
2010 LET il=LEN i$
2020 LET id=0
2030 FOR j=1 TO il
2040 LET j$=i$(j)
2050 IF j$=CHR$ 46 THEN  LET id=L
EN i$-j
2060 NEXT j
2070 LET X$=STR$ INT (ABS xi*10^i
d)
2080 LET xd=LEN x$-id
2090 FOR j=1 TO -xd
2100 LET x$="0"+x$
2110 NEXT j
2120 LET xd=LEN x$-id
2130 IF id>0 THEN  LET x$=x$( TO
xd)+"."+x$(xd+1 TO )
2140 FOR j=1 TO il-LEN x$
2150 LET x$=CHR$ 32+x$
2160 NEXT j
2170 RETURN
2180 STOP
3000 SAVE "prtusing" LINE 10
3010 STOP
4000 IF t$(LEN t$-1)="." THEN  LE
T t$=t$+"0"
4020 RETURN
```

```
  10 REM How Tall?
  15 REM Some small alts made to
     Dick's program for display
    purposes...Jack Armstrong
  20 PRINT  PAPER 5; INK 1;AT 0,5
;" How tall will you be? "
  30 PRINT : PRINT "To work out y
our final height I will need to k
now your age, yoursex and your pr
esent height."
  40 PRINT : PRINT "Firstly, what
 is your age?       (Between 2 an
d 17 only)"
  50 PRINT : PRINT "Please enter
the YEARS only..."
  60 INPUT a
  70 BEEP .05,20
  80 IF a<2 OR a>17 THEN  GO TO 6
0
  90 PRINT AT 9,30;a
 100 PAUSE 60
 110 PRINT : PRINT "Now the MONTH
S please..."
 120 INPUT d
 130 BEEP .05,30
 140 IF d<1 OR d>=12 THEN  GO TO
120
 150 PRINT AT 11,24;d
 160 LET e=a+(d/12)
 170 PAUSE 60
 180 PRINT : PRINT "Now enter ""g
"" if you are a girl  or ""b"" if
 you're a boy..."
 190 POKE 23658,0: INPUT s$
 200 BEEP .05,20
 210 IF s$<>"g" AND s$<>"b" THEN
 GO TO 190
 220 PRINT AT 14,26;("G" AND s$="
g")+("B" AND s$="b")
 230 IF s$="b" THEN  GO SUB 390
 240 IF s$="g" THEN  GO SUB 470
 250 PAUSE 60
 260 PRINT : PRINT "Lastly, pleas
e enter your heightin inches..."
 270 INPUT h
 280 BEEP .05,15
 290 PRINT AT 17,12;h
 300 LET y=(m*e)+c
 310 LET fh=h*100/y
 320 LET fh=(INT ((fh+.05)*10))/1
0
 330 LET ft=INT (fh/12): LET in=f
h-(ft*12)
 340 PRINT : PRINT "Your final he
ight will be..."
 350 PAUSE 60: FOR z=1 TO 10: BEE
P .02,z*4: NEXT z
 360 PRINT  PAPER 5: INK 1;AT 21,
6;" ";ft;" FEET, ";in;" INCHES "
 365 PAUSE 120
 370 PRINT #1; PAPER 5; INK 2;"PR
ESS ANY KEY TO RE-RUN PROGRAM "
 380 PAUSE 0: CLS : GO TO 20
 390 IF A=2 OR A=3 THEN  LET M=4.
26: LET C=40.99
 400 IF A>3 AND A<7 THEN  LET M=3
.64: LET C=43.48
 410 IF A>6 AND A<13 THEN  LET M=
3.05: LET C=47.59
 420 IF A=13 OR A=14 THEN  LET M=
4.4: LET C=30.03
 430 IF A=15 THEN  LET M=2.2: LET
 C=63.1
 440 IF A=16 THEN  LET M=1: LET C
=82.3
 450 IF A=17 THEN  LET M=.5: LET
C=90.8
 460 RETURN
 470 IF A=2 OR A=3 THEN  LET M=4.
53: LET C=43.59
 480 IF A=4 OR A=5 THEN  LET M=4.
25: LET C=44.85
 490 IF A>5 AND A<11 THEN  LET M=
3.57: LET C=48.88
 500 STOP
 510 IF A=12 THEN  LET M=3.6: LET
 C=49.7
 520 IF A=13 THEN  LET M=1.8: LET
C=73.1
 530 IF A=14 THEN  LET M=.8: LET
C=87.1
 540 IF A=15 THEN  LET M=.5: LET
C=91.6
 550 IF A=16 THEN  LET M=.4: LET
C=93.2
 560 IF A=17 THEN  LET M=0: LET C
=100
 570 RETURN
1000 REM
1010 SAVE "howtal" LINE 10
```

MAKE A WAMPUM CRATE

by: Dick Wagner

For readers who wish to reproduce this mystical crate, the following program will reproduce it. The layout is x=0 and y=0 for the bottom left corner. To relocate the image on the screen, such as x=50 and y=5, just add these values to each PLOT x and y coordinate. Remember that PLOT is absolute, starting at the bottom left corner of the screen at 0,0 while DRAW starts at a point defined by PLOT, which means that it is relative.



```
 10 REM data for CRATE
 100 PLOT 0,45: DRAW 19,19: DRAW
3,0: DRAW 0,-3
 105 PLOT 22,64: DRAW -19,-19: DR
AW -3,0
 110 PLOT 3,45: DRAW 0,-3: DRAW 1
9,19
 115 PLOT 27,42: DRAW 0,3: DRAW 1
9,19: DRAW 3,0: DRAW 0,-6: DRAW -
2,-3
 120 PLOT 49,64: DRAW -19,-19: DR
AW -3,0: PLOT 30,45: DRAW 0,-3: D
RAW 19,19
 125 PLOT 0,45: DRAW 0,-3: DRAW 1
5,0: PLOT 20,42: DRAW 10,0
 130 PLOT 5,44: DRAW 10.0: PLOT 2
0,44: DRAW 7,0
 135 PLOT 0,42: DRAW 0,-3: DRAW 1
5,0: PLOT 20,39: DRAW 10,0
 140 PLOT 22,61: DRAW 21,0: PLOT
20,59: DRAW 21,0: PLOT 20,56: DRA
W 18,0
 145 PLOT 18,18: DRAW -3,0: DRAW
0,36: PLOT 18,18: DRAW 0,39: PLOT
 20,20: DRAW 0,39
 150 PLOT 0,3: DRAW 0,3: DRAW 30,
0: DRAW 0,-3: DRAW -30,0
 155 PLOT 0,6: DRAW 2,2: DRAW 30,
0: DRAW -2,-2
 160 PLOT 2,8: DRAW 0,31: PLOT 5,
8: DRAW 0,31: PLOT 7,8: DRAW 0,31

 165 PLOT 0,3: DRAW 0,-3: DRAW 3,
0: DRAW 0,3
 170 PLOT 3,0: DRAW 5,3: PLOT 7,1
0: DRAW 8,8: PLOT 12,12: PLOT 11,
8: DRAW 11,11
 175 PLOT 27,3: DRAW 0,-3: DRAW 3
,0: DRAW 0,3: DRAW 19,19: DRAW 0,
-3: DRAW -19,-19
 180 PLOT 34,10: DRAW 8,8: PLOT 4
7,25: DRAW 2,0: DRAW 0,-3: PLOT 4
9,25: DRAW -2,-2
 185 PLOT 42,18: DRAW 0,36: PLOT
45,18: DRAW 0,38: PLOT 47,20: DRA
W 0,38
 190 PLOT 20,25: DRAW 22,0: PLOT
20,23: DRAW 22,0: PLOT 20,20: DRA
W 22,0
 195 PLOT 29,8: DRAW 0,12: PLOT 2
9,25: DRAW 0,14: PLOT 32,8: DRAW
0,12: PLOT 32,25: DRAW 0,19: PLOT
 34,7: DRAW 0,13: PLOT 34,25: DRA
W 0,21
 200 PLOT 30,42: DRAW 0,-3: DRAW
2,2
9998 STOP
9999 RANDOMIZE USR 100: SAVE "cra
te.B1" LINE 1
```

REACTION TIME

This short program will tell you how long it takes for you to press the letter "a" after "NOW" flashes some place on the screen.

```
 100 REM
 110 REM Reaction time
 120 PRINT :  PRINT  INK 2; PAPER
 5;" How many seconds will you ta
ke  to press the letter ""a"" aft
er              ""NOW"" Appears?
    "
 130 FOR c=1 TO RND*400: NEXT c
 140 PRINT AT RND*16+5,RND*25; IN
K 1; PAPER 6;"NOW"
 150 POKE 23672,0: POKE 23673,0
 160 IF INKEY$="" THEN  GO TO 160

 170 IF INKEY$<>"a" THEN  GO TO 1
70
 180 IF INKEY$="a" THEN  GO TO 20
0
 190 STOP
 200 PRINT AT 5,5;(PEEK 23672+256
*PEEK 23673)/60;" Seconds."
 210 PAUSE 100: CLS : RUN
 220 STOP
 230 REM Test by Jack Armstrong
    My best time is .2 seconds
 250 SAVE "reaction"
```

Try this one and you will see how many FRAMES there are in 1000 loops. I get 269 FRAMES.

```
 10  POKE 23672,0: POKE 23673,0
 20  FOR T=1 TO 1000: NEXT T
 30 PRINT PEEK 23672+256* PEEK
 23673
```

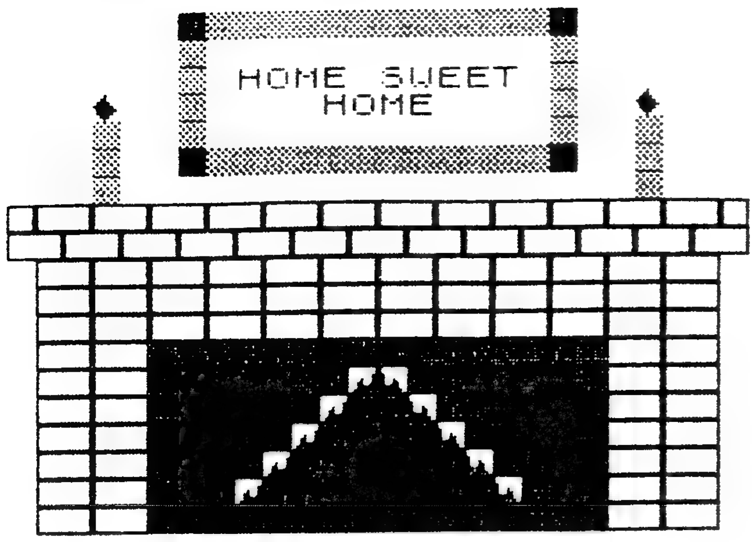
Change line 30 to divide the result by 60, as in line 230 and you will get 4.4833 seconds, or just divide the 269 by 60. Line 10 sets the FRAMES counter to 0 and line 30 reads the number of FRAMES IN 1000 loops. The divide by 60 changes the count to seconds as there are 60 FRAMES in a second. FRAMES are sent to the monitor by the computer at the rate of 60 per second. There are 2 sets of raster lines on the monitor screen, the first set tracing every other line and the second tracing in between. While you do not see the action, the first set is 262 lines and the second set is 262 lines. The time for one set of lines is 1/60 second.

SIMPLE AMORTIZATION

by: R. E. Gerow

We are pleased to see some of out long-time members coming up with ways to do "their own thing" and being willing to share this with others. We present you now with a short program submitted to us by one of our members, R. E. Gerow, of Oregon City. We hope that someone will give it a try and maybe even get some use out of it. It is a mortgage amortization program in basic that is easily modifiable. If you have any questions regarding the program, please inquire by mail at the address on the back of this newslatter.

```
  1 REM by R.E.Gerow
   5 CLS
  10 PRINT "PRINCIPLE IN $?": INP
UT a
  15 PRINT "$";a
  20 PRINT "% INTEREST?": INPUT i
i
  25 PRINT ii;"%"
  26 LET i=ii/100
  30 PRINT "LENGTH IN YEARS?": IN
PUT y
  35 PRINT y
  40 LET p=a*i/12/(1-(1+i/12)^-(1
2*y))
  50 PRINT "MONTHLY PAYMENT IS ";
: PRINT "$";INT (p+.5);".00"
  60 LET pp=p*y*12-a
  70 PRINT "TOTAL INTREST PAID ";
: PRINT "$";INT ((pp+.005)*100)/1
00
  75 PRINT
  80 PRINT "MONTHLY PAYMENT round
ed to the  nearest dollar figure.
"
9030 PAUSE 0: IF INKEY$="N" OR IN
KEY$="n" THEN  CLS : GO TO 9991
9900 PAUSE 0: CLS
9920 PRINT "FINISHED? Y/N"
9940 IF INKEY$="Y" OR INKEY$="y"
THEN  GO TO 9993
9991 GO TO 5
9993 RANDOMIZE USR 100: NEW
9999 RANDOMIZE USR 100: SAVE "amo
r.B1" LINE 5
```



```
   5 REM Fireplace designed by
      Jack Armstrong 12/84. Idea
      from a Family Computing
      program but, this is all
      original.
  10 BORDER 6: PAPER 5: INK 0: C
LS : GO SUB 500
  15 PRINT PAPER 2;AT 9,3;"□□□
□□□□□□□□□□□"; AT 10,3;"
□□□□□□□□□□□□□": REM
 Graphic D,BC(12 each),D. Graphi
c BC (13 each)
  20 FOR k=11 TO 13
  25 PRINT PAPER 2;AT k,4;"□□□
□□□□□□□□□": REM Graphi
c BC (12 each)
  30 NEXT k
  35 FOR k=14 TO 21
  40 PRINT PAPER 2;AT k,4;"□□"
;AT k,24;"□□": REM Graphic BCB
C. Graphic BCBC
  45 PRINT PAPER 1; INK 1;AT k,8
;"████████████████": REM Graphic
█
  50 NEXT k
  55 FOR k=6 TO 8
  60 PRINT PAPER 7; INK 6;AT k,6
;"▒";AT k,25;"▒": REM Graphic A
  65 NEXT k
  70 PRINT AT 19,9;"█";AT 19,22;
"█";AT 20,10;"████████████";AT 2
1,12;"█";AT 21,19;"█"
  75 FOR k=1 TO 4e4: REM Make
      this loop any length you
      want, this one is infinite
  80 PRINT PAPER 5; INK 2;AT 5,6
;"◆";AT 5,25;"◆": REM Graphic G
  82 PAUSE 10
  85 PRINT PAPER 5; INK 6;AT 5,6
;"◆";AT 5,25;"◆": REM Graphic G
  87 PAUSE 10
  90 PRINT PAPER 5; INK 2;AT 5,6
;"◆";AT 5,25;"◆": REM Graphic G
  95 PAUSE 10
 105 PRINT PAPER 1; INK 2;AT 15,
15;"▲▲";AT 16,14;"▲██▲";AT 17,13
;"▲████▲";AT 18,12;"▲██████▲";AT
 19,11;"▲████████▲": REM Graphic
E,F & █
 106 PAUSE 1
 115 PRINT PAPER 1; INK 6;AT 15,
15;"▲▲";AT 16,14;"▲██▲";AT 17,13
;"▲████▲";AT 18,12;"▲██████▲";AT
 19,11;"▲████████▲"
 116 PAUSE 1
 120 PRINT PAPER 1; INK 2;AT 15,
15;"▲▲";AT 16,14;"▲██▲";AT 17,13
;"▲████▲";AT 18,12;"▲██████▲";AT
 19,11;"▲████████▲"
 125 PAUSE 7
 130 IF k>=30 THEN PRINT PAPER 7
; INK 2; FLASH 1;AT 0,4;"MERRY C
HRISTMAS TO ALL!!"
 145 REM Graphic █ & A
 100 PRINT PAPER 6;AT 2,9;"█▒▒▒▒
▒▒▒▒▒▒▒▒█";AT 3,9;"▒
▒";AT 4,9;"▒ HOME SWEET ▒";AT 5,
9;"▒     HOME     ▒";AT 6,9;"▒
            ▒";AT 7,9;"█▒▒▒▒▒▒▒▒▒▒▒
█"
 490 NEXT k
 495 STOP
 500 REM Routine to POKE decimal
      numbers into memory creat-
      ing graphic characters
 503 RESTORE
 505 FOR a=USR "a" TO  USR "g"+7
 510 READ user: POKE a,user: NEX
T a
 512 REM DATA for Graphics
         A B C D E F & G
 513 REM ▒ [ ] □ ▲ ▲   ◆
 515 DATA 85,170,85,170,85,170,8
5,170
 520 DATA 255,128,128,128,128,12
8,128,255
 525 DATA 255,1,1,1,1,1,1,255
 530 DATA 255,129,129,129,129,12
9,129,255
 535 DATA 1,1,3,19,51,63,127,255
 540 DATA 128,128,192,200,204,25
2,254,255
 545 DATA 16,48,120,252,126,124,
56,16
 570 RETURN
 990 STOP
1000 REM
1010
1020 SAVE "fireplace" LINE 10
```

TRY THIS

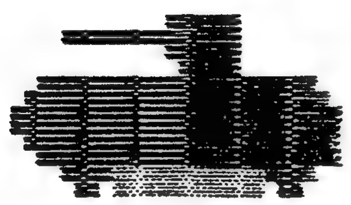
```
  10  INK 0: FOR a=0 TO 10*PI STE
P PI/20
  20 PLOT 127+(2*a)*SIN a,87+(2*a
)*COS a
  30 NEXT a
  40 INK 7
  45 PAUSE 60
  50 FOR a=10*PI TO 0 STEP -PI/20

  60 PLOT 127+(2*a)*SIN a,87+(2*a
)*COS a
  70 NEXT a
  80 GO TO 10
  90 STOP
 100 SAVE "tryths" LINE 10
```

THE OLD SHELL GAME

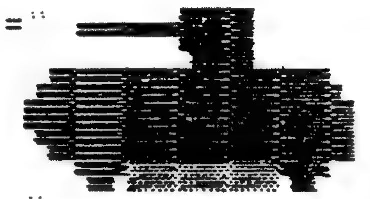
```
        ©1985 JACK ARMSTRONG
  10 BORDER 5: INK 1: PAPER 6: C
LS : CLEAR 58999: RANDOMIZE USR
100: LOAD "zt.C1"CODE : RANDOMIZ
E USR 59206: PRINT #4;" ": LET M
=0
  15 DIM S$(3): DIM P$(1): GO SU
B 20: GO TO 50
  20 FOR k=-5 TO 7: BEEP .05,k:
NEXT k: CLS : PRINT INK 2;AT 7,6
;"████████████████"
  30 PRINT FLASH 1;AT 8,6;" THE
OLD SHELL GAME "
  40 PRINT INK 2;AT 9,6;"█████
████████████": PRINT #4;"thee o
wld shell gaymmh "
  45 PAUSE 60: RETURN
  50 GO SUB 3030
  60 GO SUB 3000
  70 FOR K=1 TO 3: LET S$(K)=CHR
$ (K+143): NEXT K: LET P$=CHR$ 1
47
  80 PRINT     "Hi There, My na
me is"''"Tim Sinclair": POKE 236
58,0
  85 PRINT #4;"hye there   mye n
aymm izz  timm  ssin clayrr"
  90 PRINT ''  "What is your na
me?"
  95 PRINT #4;"whawtt izz yore n
aymm": INPUT N$: PAUSE 60: CLS
 100 PRINT ''"Well, now...";n$
 105 PRINT #4;"Well nauw  ";N$
 110 PRINT '"Do you, by any chan
ce, have a"
 115 PRINT #4;"Doo yoo  bye ehny
 chance  have uh"
 120 PRINT "bit of gambling blo
od in you?"
 125 PRINT #4;" bit  uhv  gahmbl
ing blud in yoo"
 130 PRINT ''"Input your answer:
(Yes) or (No)": POKE 23658,8
 135 PAUSE 30: PRINT #4;"Innputt
 yo rr  anser  Yess  ore  Noe "
 PAUSE 0: LET O$=INKEY$
 140 IF O$<>"Y" AND O$<>"N" THEN
 CLS : GO TO 130
 150 IF O$=CHR$ 78 THEN GO TO 61
0
 160 CLS : PRINT '''"Well now,
";n$;" they call this:"
 165 PRINT #4;"Well  nauw ";n$;"
 thay call thiss:": PAUSE 60
 170 CLS : GO SUB 20
 180 PAUSE 60: CLS
 190 PRINT ''''"Here's the deal
   ";n$
 195 PRINT #4;"Heerzz thuh deel
 ";n$
 200 PRINT  "I have these three
shells...": TAB 10; INK 2;S$;" "
;S$;" ";S$
 205 PRINT #4;"Eye have theez th
ree shells"
 210 PRINT  "And I have this lit
tle pea...": INK 4;P$
 215 PRINT #4;"and  eye have thi
ss  little pee": PAUSE 60
 220 GO SUB 1000
 230 PAUSE 60: CLS
 240 PRINT ''"Here's the dope "
;n$;"..."
 245 PRINT #4;" heerzz thuh dope
 ";n$
 250 PRINT '"I'll put the pea un
der a shell."
 255 PRINT #4;"  eyell putt  thu
h pee  under uh shell"
 260 PRINT '"Mix them up...Then
you guess..."
 265 PRINT #4;"mmix them up   th
en yoo guess  "
 270 PRINT '"Which shell is the
pea under... ": 'TAB 11;"1 - 2 o
r 3"
 275 PRINT #4;"which shell  izz
thuh pee under   wun  too   ore
three     "
 280 PRINT '"Just to make things
 interesting-" "Let's make a li
ttle wager on it.": LET m=10
 285 PRINT #4;"just  too mayk  t
hingz intresting  letz mayk   ay
e little wayger on it   "
 290 PRINT ''"Press ENTER to con
tinue..."
 295 PRINT #4;"press  enter  too
 continue": PAUSE 0: CLS : PRINT
 '"How much do you want to bet t
hat"
 297 PRINT #4;"howh much  doo yo
o wont too bet  that"
 300 PRINT  "You can guess corre
ctly?"
 305 PRINT #4;"yoo can guess  co
rrectly  "
 310 PRINT ''"Since we are frien
ds here, let's"
 315 PAUSE 10: PRINT #4;"since w
ee  are frendz heer  letz "
 320 PRINT "make some limits-sa
y you have"''"$10.00 and you can
 bet any even"
 325 PRINT #4;"mayke  sum lihmit
z  say yoo have ten dollurz   an
d yoo can bet  eny even"
 330 PRINT '"amount from $1 to $
10.00 as long"''"as you have the
 money to bet."
 335 PRINT #4;"uhmount  from  wu
n too ten dollurz  az long az yo
o have thuh munnee too bet"
 340 PRINT ''    "Press ENTER to
continue..."
 345 PRINT #4;"press enter  too
continue ": PAUSE 0: CLS : PRINT
 ''''"PLACE YOUR BET. Please en
ter the" "number only. Don't us
e the ($)"
 347 PRINT #4;"playce yore bet
pleez enter thuh numbur only  do
e nt  uze "
 350 PRINT  "dollar sign-just th
e number."
 355 PRINT #4; thuh dollur sine
 just thuh number"
 360 PRINT AT 20,0; INK 2; PAPER
 7;n$;" ": PRINT AT 21,0;"You ha
ve $"; FLASH 1;m; FLASH 0;" in y
our poke."
 365 PRINT #4; chek thuh skreen
 fore thuh munnee  inn yore poke
": INPUT ;bet: PAUSE 60: CLS
 370 IF bet>m THEN GO TO 410
 380 IF bet>10 THEN GO TO 410
 390 LET m=m-bet
 400 IF bet>=1 AND bet<=10 THEN
GO TO 430
 410 PAUSE 60: CLS : PRINT ''
"Come, come, Sport-I'm no sucke
r"
```

```
 415 PRINT #4;"kum kum  sport  e
yemh noe sukker "
 420 PRINT ''"Quit trying to con
me-Make your bet!"
 425 PRINT #4;"qwit trying to ka
wnn me  mayke yore bet": IF bet>
M OR bet>10 THEN PRINT '"You can
't bet more than $10.00  "'"or m
ore than is in your poke.": PRIN
T #4;" yoo cannt bet  more than
tenn dollurz  ore more than iz i
n yore poke": LET bet=0: PAUSE 1
20: CLS : GO TO 340
 430 PRINT ''      "O.K., Sport...H
ere we go..."
 435 PRINT #4;"owe kaye  sport
heer wee goe"
 440 GO SUB 20: GO SUB 1000
 450 LET P=INT (RND*3)+1
 460 PRINT ''    "Well, Now, ";n$;
"..."
 465 PRINT #4;"well  nauw  ";n$
 470 PRINT '"Where's the Pea?"
 475 PRINT #4;"wherez thuh pee
whawtz  yore gess": INPUT "What'
s your guess: ";g
 480 CLS : IF g<1 OR g>3 THEN GO
 TO 460
 490 PRINT ''   "O.K., Sport, gl
ad you made that"'"Choice-Let's
 see now..."
 495 PRINT #4;"owe  kaye  sport
 glad yoo mayde that choice  let
z see nauw "
 500 IF g=p THEN LET m=m+2*bet
 510 IF g<>p THEN GO TO 670
 520 PAUSE 60: CLS : GO SUB 800:
 PAUSE 60
 530 PRINT ''"How about that, Sp
ort-You made a"
 535 PRINT #4;"how about that  s
port  yoo made aye"
 540 PRINT '"good guess...Now yo
u have $";m;"."
 545 PRINT #4;"good guess  now y
oo have too chek thuh skreen for
e what yoo have inn yore poke "
 550 PRINT '"Want to try again?
If you feel"
 555 PRINT #4;" waw nt too trye
again   iff yoo feel"
 560 PRINT '"Lucky, Input (Y)es
or if you are"
 565 PRINT #4;"lucky  innputt wy
e  ore  if yoo are"
 570 PRINT '"just a piker-Input
N(o)": PRINT #4;"just aye piker
 innputt enn": INPUT f$: PAUSE 6
0: CLS
 580 IF f$="N" THEN GO TO 610
 590 IF f$="Y" THEN GO TO 340
 600 IF f$<>"Y" OR f$<>"N" THEN
PRINT  "Hey, Sport, Y or N only.
": PRINT #4;"hay sport  wye ore
enn only": PAUSE 120: CLS : GO T
O 550
 610 PAUSE 60: CLS : PRINT ''''
"O.K., Sport, No hard feelings..
."
 615 PRINT #4;"owe kay  sport  n
oe hard feelings "
 620 PRINT '"See you around, ";n
$;"..."
 625 PRINT #4;"see yoo around ";
n$
 630 PRINT  "You had $";m;" left
..."
 635 PRINT #4;"yoo haff to chek
thuh skreen fore what yoo haff l
eft"
 640 PRINT '"If you'd like to tr
y again,"
 645 PRINT #4;"iff yooud like to
o try again "
 650 PRINT  "Press R for a re-ru
n...": PRINT #4;"press  aw urr
fore uh ree run": PAUSE 0: IF IN
KEY$="R" THEN CLS : LET M=10: GO
 TO 340
 660 PAUSE 60: CLS : PRINT #4;"
sow long  sukkerr "  STOP
 670 PAUSE 60: CLS : GO SUB 800
 680 PAUSE 60: CLS : PRINT ''''
"Sorry, Sport-you missed that on
e"
 685 PRINT #4;"sorry  sport  yoo
 missed that wun  chek thuh skre
en fore how much yoo have left"
 690 PRINT '"You now have $";m;"
 left."
 700 IF M<1 THEN GO TO 740
 710 PRINT '      "If you want to
try again Press"'"Y(es), if not
 Press N(o)."
 715 PRINT #4;"iff yoo waw nt  t
oo trye again   press wye   iff
not   press  enn"
 720 PAUSE 0: IF INKEY$="Y" THEN
 GO TO 340
 730 IF INKEY$="N" THEN GO TO 61
0
 740 PAUSE 60: CLS : PRINT ''
"If you'd like to play again..."
 745 PRINT #4;"iff youwd like to
 play again"
 750 PRINT '"Press R to Re-Run..
."
 752 PRINT #4;"press  aw ur  to
ree run"
 755 PAUSE 0: IF INKEY$=CHR$ 82
THEN CLS : LET M=10: GO TO 340
 760 PAUSE 10: PRINT #4;" good
bye": STOP
 800 PAUSE 60: CLS : PRINT ''
"The Pea was under..."
 805 PRINT #4;"thuh pee  wuz und
er "
 810 PRINT INVERSE 1;AT 8,(8 AND
 P=1)+(16 AND P=2)+(24 AND P=3);
P
 820 PRINT INK 2;AT 10,3;"    AB
C     ABC     ABC";AT 10,(8 AND
P=1)+(16 AND P=2)+(24 AND P=3);
INK 4;P$
 825 PRINT #4;("wun" AND p=1)+("
too" AND p=2)+("three" AND p=3)
 830 PAUSE 60: RETURN
 999 STOP
1000 REM
1010 CLS
1020 LET v=10: LET h=5: FOR l=1
TO 3
1030 IF l=2 THEN LET h=h+10
1040 IF l=3 THEN LET h=h+10
1050 PRINT INK 2;AT v,h;s$
1060 NEXT l
1070 PRINT AT 8,6;"1";AT 8,16;"2
";AT 8,26;"3"
1080 LET D=10: FOR L=1 TO 14
1090 LET X=(RND*30)+1: IF X<=10
THEN LET X=6
1100 IF X>10 AND X<=20 THEN LET
X=16
1110 IF X>20 THEN LET X=26
1120 PRINT INK 4;AT D,X;P$: PAUS
E 10
1130 PRINT INK 2;AT D,X;S$(2): P
AUSE 5
1140 BEEP .01,.05: NEXT L
1150 RETURN
3000 REM
```

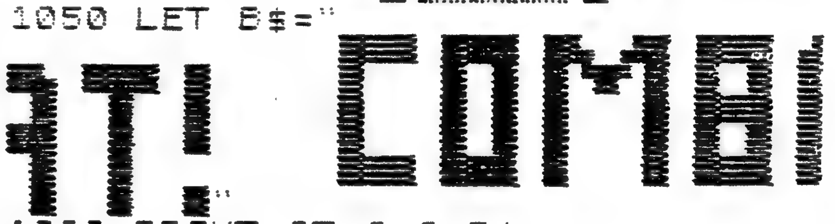
```
3010 CLS : PRINT "To Play T
his Game...","Follow Instructio
ns Carefully...","Press ENTER a
fter each Input or just press th
e key required."; TAB 11;"Good L
uck!"
3015 PRINT #4;"too play  this ga
yme  follow instrukshunz care fu
llee  press  enter  afftur each
 innputt  ore just press thuh ke
e ree kwyerrd    good  lukk"
3020 PRINT AT 18,0;"Press ENTER
to Start..."
3025 PRINT #4;"press entur  too
start": PAUSE 0: CLS : RETURN
3030 RESTORE 3060
3040 FOR a=USR "a" TO USR "d"+7
3050 READ user: POKE a,user: NEX
T a: RETURN
3060 REM DATA 0,3,12,16,32,64,12
8,255
3065 DATA 0,3,12,16,32,64,128,25
5
3070 REM DATA 60,195,0,0,0,0,0,2
55
3075 DATA 255,0,0,0,0,0,0,255
3080 REM DATA 0,192,48,8,4,2,1,2
55
3085 DATA 0,192,48,8,4,2,1,255
3090 DATA 24,126,250,247,239,94,
126,24
3095 STOP
4000 CLS
4010 RANDOMIZE USR 100: SAVE "sh
ell.B1" LINE 10
4020 RANDOMIZE USR 100: SAVE "zt
.C1"CODE 59200,6000
```



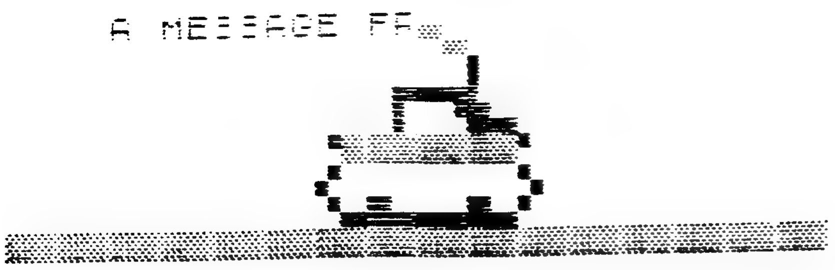
```
  20 REM BY JACK ARMSTRONG
  30 LET SCORE=0
  40 FOR J=1 TO 10
  50 GO SUB 1000
  60 PRINT AT 0,3;"ENTER A NUMBE
R FROM 1 TO 10";AT 1,11;"GO NUMB
ER ";J
  70 PRINT AT 20,0;"YOU NEED ";5
-SCORE;" TO WIN": INPUT A
  80 IF A<1 OR A>10 THEN GO TO 7
0
  90 PRINT AT 10,0;"YOUR NUMBER
IS ";A;AT 12,6;"SCORE IS ";SCORE
 100 FOR G=1 TO 4
 110 LET B=INT (RND*10)+1
 120 PRINT AT 6,7;"VOLLEY #";G;A
T 8,9;"VALUE ";B;" "
 130 IF B=A THEN GO TO 150
 135 PAUSE 40
 140 NEXT G
 150 IF A=B THEN LET SCORE=SCORE
+1
 160 IF A=B THEN PRINT AT 14,7;"
WELL DONE"
 170 IF A<>B THEN PRINT AT 14,8;
"BAD LUCK "
 180 PRINT AT 12,6;"SCORE IS ";S
CORE
 190 IF SCORE=5 THEN GO TO 250
 200 FOR T=1 TO 20
 205 PAUSE 5
 210 NEXT T
 220 CLS
 230 NEXT J
 240 PRINT "THE GAME IS OVER AND
YOU ONLY  SCORED ";SCORE
 250 PRINT "YOUR RATING IS ";SCO
RE/.05;" PERCENT"
 260 GO TO 305
 270 PAUSE 60
 280 PRINT "YOU DID IT"
 290 REM SCROLL goes here. The
     next line works on the 2068
 295 RANDOMIZE USR 2361
 300 PRINT "YOU WIN"
 305 PAUSE 120
 310 RANDOMIZE USR 2361
 320 PRINT "ANOTHER TRY? PRESS "
"RUN""，""ENTER"""
 330 STOP
1000 REM This was originally for
     the 1000 now updated for
     the 2068
1005 RESTORE
1010 FOR k=USR "a" TO USR "a"+7
1020 READ udg: POKE k,udg: NEXT
k
1030 DATA 85,170,85,170,85,170,8
5,170
1035 BORDER 6: PAPER 5: INK 0: C
LS
1040 LET A$="
```



```
"
1050 LET B$="
```



```
"
1060 PRINT AT 3,3;B$
1070 PRINT AT 11,10;A$
1075 PAUSE 100: CLS
1080 RETURN
1090 STOP
1100 SAVE "COMBAT"
1110 GO TO 10
1120 STOP
1200 SAVE "combat" LINE 10
```



```
  10 REM A SCROLLING XMAS
     MESSAGE
  15 BORDER 6: PAPER 5: INK 1: C
LS : REM THIS IS FOR THE 2068
  20 LET B=0
  30 LET O=1
  40 LET L=0
  50 LET A$="* * * "
  60 LET A$=A$+"SEASON'S GREETIN
GS FROM THE CLACKAMAS COMPUTER A
PPLIED TRAINING SOCIETY."
  70 LET A$=A$+" MAY YOUR PROGRA
MS ALL BE KEYED IN CORRECTLY AND
 YOUR SCREEN DISPLAYS GLOW WITH
PROGRAMMING WISDOM."
  80 LET A$=A$+" * * *"
  90 LET B$="* * * "
 100 LET B$=B$+"CCATS IS PROUD T
O BE A PART OF THE T/S ACTION. W
E ARE NOW INTO OUR 9TH YEAR AS A
 GROUP AND ARE LOOKING FORWARD "
 110 LET B$=B$+"TO NEW PROGRAMS,
 NEW HARDWARE AND CONTINUED EXPA
NSION OF T/S MAGAZINES. "
 120 LET B$=B$+"WE WANT EVERY SU
PPLIER OF OUR NEEDS TO KNOW THAT
 THEIR EFFORTS TO SATISFY ARE TR
ULY APPRECIATED. * * *"
 130 LET C$="▒"
 140 REM SLOW goes here for 1000
 150 PRINT "CCATS USER GROUP - BU
LLETIN BOARD"
 160 FOR N=0 TO 30
 170 LET C$=C$+"▒"
 180 NEXT N
 190 PRINT AT 5,0;C$;AT 9,0;C$;A
T 6,0;"█";AT 7,0;"█";AT 8,0;"█";
AT 6,31;"█";AT 7,31;"█";AT 8,31;
"█"
 200 PRINT AT 15,0;C$;AT 19,0;C$
;AT 16,0;"█";AT 17,0;"█";AT 18,0
;"█";AT 16,31;"█";AT 17,31;"█";A
T 18,31;"█"
 210 LET C$="
"
 220 GO SUB 330
 230 LET L=LEN D$
 240 FOR B=1 TO L
 250 IF B>L-27 THEN GO TO 310
 260 PRINT AT A,2;D$(B TO B+27);
 265 PAUSE 7: REM THIS IS FOR
     THE 2068
 270 NEXT B
 280 GO SUB 330
 290 LET L=LEN D$
 300 GO TO 240
 310 PRINT AT A,2;D$(B TO L);
 315 PAUSE 7: REM THIS IS FOR
     THE 2068
 320 GO TO 270
 330 IF O=1 THEN GO TO 380
 340 LET D$=C$+B$+" "
 350 LET A=17
 360 LET O=1
 370 RETURN
 380 LET D$=C$+A$+" "
 390 LET A=7
 400 LET O=0
 410 RETURN
 420 REM
 430 STOP
 500 REM
 510 SAVE "bulitnbrd" LINE 10
```

```
  10 REM I USED A SCROLLING
     MESSAGE: "[illegible]
     [illegible]
     [illegible]" TO MAKE A DISPLAY
     THAT LOOKS GOOD. TYPE UP
     TO THE END OF EACH LINE,
     NO EXTRA SPACES.
  20 REM LINES 170 THRU 210 PRO-
     VIDE THE SCROLL SAME LEVEL
     AS THE EXHAUST AND START AT
     CORRECT POINT.
  30 REM GRAPHIC CHARACTERS ARE
     SHIFTED 9 & SHIFTED #'S &
     LETTERS IN THIS ORDER: D S
     3 5 6 6 6 5 2 SPACE 6 2 1 G
     G G G G G G T T 2 D G D D D
     G D.
  40 REM IN LINE 170 USE H 32
     TIMES-DON'T COUNT, JUST
     TYPE UNTIL THE ENDS ALINE
  45 GO SUB 400
  50 PRINT ,,"INPUT MESSAGE WHEN
 PROMPTED BY: ""L"""
  60 PAUSE 100
  70 CLS
  80 PRINT AT 8,16;"▒"
  90 PRINT AT 9,17;"▒"
 100 PRINT AT 9,18;"▖"
 110 PRINT AT 10,18;"▌"
 120 PRINT AT 10,15;"▄";AT 10,16
;"▄";AT 10,17;"▄";
 130 PRINT AT 11,15;"▌";AT 11,17
;"▘";AT 11,18;"█";AT 11,19;"▄"
 140 PRINT AT 12,12;"▘";AT 12,20
;"▘";AT 12,13;"▒▒▒▒▒▒▒";AT 13,12
;"▞      ▚"
 150 PRINT AT 14,12;"▘";AT 14,13
;"▄";AT 14,14;"█";AT 14,15;"▄▄▄"
;AT 14,18;"█";AT 14,19;"▄";AT 14
,20;"▘"
 160 PRINT AT 15,0;"▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒▒"
 170 INPUT a$
 180 LET a$="                "+a$
 190 PRINT AT 8,0;a$( TO 16): BE
EP .009,8: PAUSE 7
 200 LET a$=a$(2 TO LEN a$)+a$(1
)
 210 GO TO 190
 220 STOP
 230 SAVE "SCROLL"
 240 RUN
 390 STOP
 400 REM These subroutines are
     only for the 2068
 410 BORDER 2: PAPER 6: INK 0: C
LS
 500 REM
 510 FOR K=USR "A" TO USR "C"+7
 520 READ UDG: POKE K,UDG
 530 NEXT K
 540 RETURN
 550 DATA 0,0,0,0,85,170,85,170
 560 DATA 85,170,85,170,0,0,0,0
 570 DATA 85,170,85,170,85,170,8
5,170
 580 STOP
 590 SAVE "scroll" LINE 10
 600 REM The above for the 2068.
     The program was designed
     for the 1000 & expanded
     here to 2068 use
 610 REM For the 1000, leave out
     the parts that apply to the
     2068
  10 BORDER 2  INK 0  PAPER 5: C
LS
  20 PRINT INVERSE 1,"      CASSE
TTE LABEL PROGRAM         "
  30 PRINT ,,"Use with printer t
o make labels "
  40 PRINT ,, PAPER 1; INK 2;"Yo
u may print 3 lines. Enter oneli
ne at a time followed by ENTER"
  45 PRINT : PRINT "Make sure yo
ur lines are no morethan 30 char
acters in length.      "
  50 PRINT PAPER 3; INK 0;AT 20,
0;"Now enter line 1:"
  60 PRINT AT 10,0;"▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄"
  70 FOR i=1 TO 3: PRINT "▌";TAB
 31;: INVERSE 1: PRINT "▌": INVE
RSE 0
  80 NEXT i
  90 PRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄"
 100 INPUT a$
 110 IF a$<>"" THEN LET x$=a$
 120 IF a$="" THEN GO TO 250
 130 IF LEN a$>30 THEN GO SUB 90
0: GO TO 100
 140 LET t=30-LEN a$
 150 PRINT AT 11,1;"
                    ";AT 11,t/2;a$
 160 REM
 170 REM
 250 PRINT PAPER 2; INK 0;AT 20,
0;"Next, please enter line 2:"
 260 INPUT b$
 270 IF b$<>"" THEN LET y$=b$
 280 IF b$="" THEN GO TO 350
 290 IF LEN b$>30 THEN GO SUB 90
0: GO TO 250
 300 PRINT AT 12,1;"
                    "
 310 LET t=30-LEN b$
 320 PRINT AT 12,t/2;b$
 330 REM
 340 REM
 350 PRINT AT 20,0;"
                   ";AT 20,0; PAPER 1
; INK 6;"Now enter last line:"
 360 INPUT c$
 370 IF c$<>"" THEN LET z$=c$
 380 IF c$="" THEN GO TO 470
 390 IF LEN c$>30 THEN GO SUB 90
0: GO TO 360
 400 PRINT AT 13,1;"
                    "
 410 LET t=30-LEN c$
 420 PRINT AT 13,t/2;c$
 430 REM
 440 REM
 450
 480 PRINT AT 19,0; INVERSE 1;""
"C"" for Copy, ""N"" for New Lab
el,            & ""S"" to STOP
     "
 490 PAUSE 0
 495 POKE 23658,0: REM for CAPS
 500 LET d$=INKEY$
 505 IF d$="s" THEN CLS : STOP
 510 IF d$="n" THEN CLS : RUN
 515 IF CODE d$=13 THEN GO TO 48
0
 520 IF d$="" THEN GO TO 480
 530 LPRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄"
 540 LET r=30-LEN x$: LET s=30-L
EN y$: LET t=30-LEN z$
 550 LPRINT "▌";AT 1,r/2;x$;TAB
31;"▌"
 560 LPRINT "▌";AT 1,s/2;y$;TAB
31;"▌"
 570 LPRINT "▌";AT 1,t/2;z$;TAB
31;"▌"
 580 LPRINT "▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄▄
▄▄▄▄▄▄▄▄▄▄▄▄▄"
 590 LPRINT : LPRINT : LPRINT
 600 GO TO 480
 900 PRINT INVERSE 1;AT 20,0;"To
o many letters please re-enter"
 910 RETURN
 950 STOP
1000
2000 REM SAVE "label" LINE 1
2010 REM RANDOMIZE USR 100: SAVE
 "label.B1" LINE 1
2020 REM use line 2010 if you
     are using Larken
3000 REM There were lines in the
original listing printed in Vol.
3, No. 1 that were "BUGS" so all
of them were taken out and REM
lines were left in their place.
3010 REM See 160,170--330,340--
430,440...they were: GO SUB 900
and GOTO 100,260,350. They were
extraneous and would not allow
the program to run properly.
3020 REM The whole program was
edited with subtle changes that
improved and shortened it. It
really needs to be totally re-
done with GOSUBs to elimimate
3030 REM redundancies, however
it could be the basis for a good
label maker. All sorts of things
could be added.
```

TASMAN POKES

by Randy Kuhn

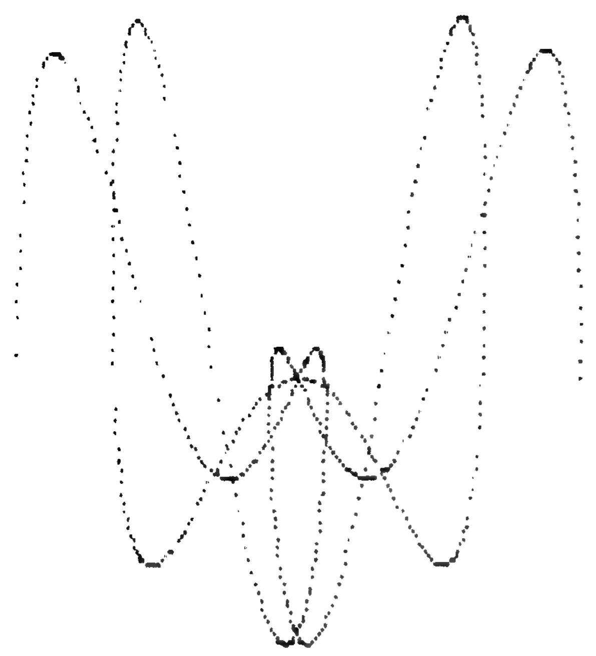
Some POKEs I have found to use with the Tasman Printer Interface with Mterm and HOT-Z are as follows:

Mterm: POKE 54445,0 and 54446,91

Hot-Z High: POKE 51803-51806,0
    51807,205 51808,0 51809,19
    51810,201

Hot-Z Low: POKE 30303,0 30304,205
    30305,0 30306,91 30307,201

For Mterm the only time it will copy is when you hit caps shift 8 and hit P for PRINT. I haven't found the one for the buffer yet. For Hot-Z it is only when you use the COPY command.



EASY KEYBOARD FIX

By Dick F. Wagner

The 2068 computer has a poorly designed space bar, as usually the right hand end does not produce a space when pressed. I have explained to users about how to make a fix by disassembling the keyboard and inserting spacers between the hinge arms and the case to slightly lower the space bar and reduce its travel. This is not easy to accomplish and requires removing the plastic keyboard overlay.

Here is an extremely easy solution to the problem that does not require entering the case , or even disconnecting the cables or plug-ins. One needs only a bright light, scissors, some note paper in weight like 3M adhesive notes, and tweezers or needle nose pliers.

Inspect the space bar arrangement with a bright light and see how it moves. With the computer on, try pressing slightly on the left end and at the same time press the right end. The repeat should come on. Turn off the computer and note the gap between the back of the case and the case, and see the two parts about 1/8 inch wide and 5/8 inch from each end that move within two slots. This is where the improvement is made. The arms are hinged about 1 inch back of he space bar.

The reason for the poor key action is that there are 2 contacts under the space bar that are too close together, plus the flexibility of the arms that do not force the key to move fully no matter where pressed. The springs under the keys are just rubber dimples that "turn inside out" when pressed. A hard button on the top of each makes the actual switch contact. This is all a 1 piece rubber assembley that seals the contacts from dirt.
The fix is simple-- just lower the space bar about 15 thousands of an inch. Cut some 1/8 inch wide strips from the note paper which is about 0.004 inch thick (don't use the adhesive end if 3M paper is used). Cut 2 lengths about 3/4 inch long and 2 lengths about 3/8 inch long. Fold the longer strips at the mid length in a tight fold. Insert a short piece in each in each folded strip. At the loose end bend very slightly in the same direction at a point about 1/16 inch from the end.

With your good light on, press the space bar fully and gripping the paper spacer (open end down and the right angle bent away from you) insert it between the space bar and tne case, directly over an arm. The slight bend on the end is to make it easier to slide the paper between the arm and the case. Slide the paper down to the bend point, leaving the folded end protruding slightly higher than the case proper. This makes it easier to retrieve the spacers if necessary. The right angle bend in the spacer keeps the spacer from slipping too far and it also keeps it from slipping sideways. If the strip ever does slip inside the case no damage can occur as all contacts are sealed.

Warning--as the rubber "springs" are not powerful try not to bend the open end of the paper spacer very much as the spacer may not flatten properly and the spacer will be effectively too thick.

Test and see the improvement. If your space bar is not fully fixed and requires excessive pressure on the end, remove the strips and add another 3/8 inch insert. It would be better to make a new set than try to work with the old set because of the bends. One user of this fix found that 4 thicknesses worked better than 3. There are differences in hardware so try a combination that works. It only takes a few minutes!

*D & *E COMPILATION BUG

TIMEMACHINE compiler users may be interested in this correction. Apparently it only shows up when compiling large programs and *D and *E are used

After the program has loaded (while the backup prompt is still on the screen), break into the program by deleting the left quote and entering STOP (Symbol Shift A). Add the following lines to BASIC.

TS 2068 Version

```
  20 FOR i = 1 TO 4
  21 READ address, n
  22 FOR j = 0 TO n-1
  23 READ byte: POKE address+j,
byte
  24 NEXT j: NEXT i

 100 DATA 26843,4
 101 DATA 205,0,130,0
 102 DATA 32070,4
 103 DATA 205,0,130,0
 104 DATA 32858,4
 105 DATA 205,71,104,0
 106 DATA 33280,12
 107 DATA 17,0,0,205,71,104,192,
237,91,241,68,201
```

Now, GOTO 8000 and ENTER to make e a new, corrected, backup copy.

SPECTRUM Version

Change lines 100-107 as follows:

```
 100 DATA 23952,4
 101 DATA 205,241,133,0
 102 DATA 29145,4
 103 DATA 205,241,133,0
 104 DATA 29906,4
 105 DATA 205,247,92,0
 106 DATA 34288,12
 107 DATA 17,0,0,205,247,92,192,
237,91,241,68,201
```

PRINTER ZERO

by Dick F. Wagner

Some large printers do not provide a "0"for zero but it is handy to have at times. The regular zero is usualy slightly different than a "O". The Panasonic 1080i printer lacks this feature but it is easily added and some other Epsom compatables should be just as easy to change. One printer design requirement is access to the printer RAM.

Access to the RAM is by ESC+y+D where ESC is LPRINT 27, y is LPRINT 121, and D is the character code location. Select the location for the same as zero or character code 48. Thus every time the zero key is pressed the printer produces the desired slashed character.

The program I save on disc is this:

```
  10 LPRINT CHR$ 27;CHR$ 121;CHR
$ 48;
  20 FOR N=1 TO 9
  30 READ a
  40 LPRINT CHR$ a;
  50 NEXT N
  60 DATA
58,68,0,138,16,162,0,6
8,184
```

FLOWER BASKET

(GRAPHIC)

By Dick F. Wagner

The Flower Basket design was found in the Mar.-Apr. 88 issue of SyncWare News as a UDG graphic developed by Dorthea Bundy. UDGs are defined horizontally, top down and binary 1 the top right corner. In printer graphics (Epson) the print pin positions are defined as vertical columns, bottom is binary 1. This excludes taking the UDG data and rotating it to get the new data. The 136 data statements were developed by making a graph paper layout and then reading the column/rows data in binary for the new printer data.

Line 1 is applicable to my Oliger printer driver and takes the place of OUT 127,n and allows the use of LPRINT. Printer commands are applicable to Epsom work alike printers. Line 230 produces a line feed. Line 240 sends READ to line 260 on negative numbers. Then P READS the the number following this number. It repeats until there are no more negative numbers then goes back to READ N. This saves duplicating DATA. In this program this procedure prints empty spaces. Lines 240, 260, and 270 is a subroutine for STRING$ available on some other computers. This program is for 8 bit printer mode and will not work on 2040 printers.

```
 10 REM  LET /p=o: POKE 23300,60
: POKE 23301,3
 100 LPRINT CHR$ 27;"1"
 200 LPRINT : NEXT K
 210 LET A$=CHR$ 27+"K"+CHR$ 34+C
HR$ 0
 220 FOR K=1 TO 4: LPRINT A$
 230 READ N: IF N=999 THEN  GO TO
280
 240  IF N<0 THEN  GO TO 260
 250 LPRINT CHR$ N;
 260 READ P: FOR J=1 TO -N: LPRIN
T CHR$ P;: NEXT J
 270 GO TO 230
 300 DATA -14,0,3,12,32,108,32,12
,3,-12,0,0,999
 310 DATA 0,3,6,6,14,6,2,34,49,10
3,31,13,1,1,255,14,23,29,22,29,25
5,0,43,14,55,59,24,2,6,14,4,6,3,0
,999
 320 DATA 224,32,0,0,14,49,64,200
,216,240,152,184,180,218,213,186,
216,240,152,184,180,218,213,186,2
45,186,21,250,213,218,20,184,216,
176,88,200,64,50,14,0,32,224,999
 330 DATA 0,-4,0,128,0,-4,0,2,6,6
2,202,62,202,62,202,62,6,2,0,-10,
0,999
9999 SAVE "FLOWER"
```

FAT CHARACTERS

by: Syd Wyncoop

This routine is only 39 bytes and can be loaded to any available free memory, such as the printer buffer.

The first four lines perform a block move and copy the ROM character set to FC00h (64512). This is as high in memory as is convenient, without overwriting the UDGs.

The next four instructions place the address of the "new" character set into the proper system variable locations.

The remaining instructions comprise a loop that moves each row (byte) of pixel information, on bit to the right. Then it ORs (adds) back in the origional bit locations. (See figure 1) The new data is then re-stored to the new character table.

There is no further action needed by you, as the routine creates and tells the system about the new character table.

To use it, LOAD the routine to any available 39 byte area and call it with a RANDOMIZE USER address. Of course, address is the address you LOADed the routine to.

Have Fun!

Listing 1

```
010003 FatChars    Ld BC, 0300h
1100FC             Ld DE, FC00h
21003D             Ld HL, 3D00h
EDB0               Ldir
2100FC             Ld HL, Fc00h
FD75FC             Ld (IY+FCh),L
FD74FD             Ld (IY+FDh),H
FD35FD             Dec    (IY+FDh)
10003              Ld BC, 0300h
7E       Loop      Ld A, (HL)
5F                 Ld E, A
CB0F               RRC A
B3                 Or E
77                 Ld (HL), A
0B                 Dec BC
23                 Inc HL
78                 Ld A, B
B1                 Or C
20F4               Jr NZ, Loop
C9                 Ret
```

Figure 1.

```
Original byte = 0 0 1 1 0 1 0 0
Rotated byte  = 0 0 0 1 1 0 1 0
ORed byte     = 0 0 1 1 1 1 1 0
```

STRING$

STRING$ is a BASIC command used in some BASIC languages and is not the same as STR$. Knowing its meaning is necessary for SINCLAIR BASIC replacement.

STRING$ is used to creat strings made up of the same character in graphics, as for borders. You define the character and the desired number of repeats.

The SINCLAIR BASIC replacement for:

```
  10 B$=STRING$(32,"-")
```

is:

```
  10 LET T$="": FOR N= TO 32: LET
T$=T$+"-": NEXT N: LET B$=T$
```

MORE ON TASWORD 2

by: Dick Wagner

At one time or another, users of TW2 have had occasion to look at or change printer codes. The menu gives this option, the TW2 programmers recognizing the needs of different printers. Perhaps the reader has changed printers, thus requiring one or more changes.

I have altered codes several times, starting with a Olivetti printer. For sure I didn't understand all that I was doing. Why did some codes in the original list include 32? Nothing was said in the meager instruction manual about this. Lucky for me, my changes didn't seem to effect the printer output, if I knew enough to look for them.

Recently I came across a letter by P. F. Green (Rotterdam) in the ZX Computing Monthly for February 1987, in the Cross Wires column in which he addressed the subject of code 32. From his discussion it appears that the code 32 is used to make the lines justify properly on the printed page.

Refering to the 2068 list of characters and codes in that manual, it will be noted that 32 is the code for "space" (we all new that, didn't we?) Where the right justify does not print properly, look at the codes being sent to the printer. If only default codes are being used, then look at these to see if the right justify can be adjusted by adding a space in one of those codes. It the reader has added a printer code to the text, say at one point where Italics is desired, then this could be the culprit.

A good test would be to delete a 32 in a code and use the new one to print a text. This should show the effect of the code 32 when right justifying.

Another TW2 improvement was published in the March-April 1985 issue of Syncware News, page 5. This was written by Duncab Teague. (It is surprising how much information can be dug out ol these old magazines.)

Several times in our old THE PLOTTER issues we have published information on how to revise the first "HELP" page of TW2. This information is a little different approach that will make it possible to edit both "HELP" pages. Why change them? Have you changed printer codes for a printer function that was not on the tabulation of codes? Have you changed printers but have left the headings unchanged? Have you made a note of the correct function as a substitute to keep track? Why not correct these changes properly and have a real nice menu?

This process takes the "HELP" pages out of the memory locations and puts them into text files so they can be edited. Then the edited files are returned to their original addresses.

#1--with TW2 loaded, STOP to get to basic and then type in the command mode (no line numbers) the following:

```
 FOR i=0 TO 1535: POKE
(33280+i), PEEK (5478+i): NEXT i
```

#2--GO TO 20 and select "y" to return to the text file. the first of the "HELP" pages in now in the text file and can be read and revised.

#3--After editing, reverse the addresses and poke the screen back to memory.

#4--The second "HELP" pages in sent to the textfile with the following line:

```
 FOR i= TO 1535: POKE (33280+i),
PEEK (56320+i): NEXT i
```

NOTE: this address is 1536 higher.

#5--Repeat as in #2 and #3. reverse the addresses and poke the screen back to the new memory location.

Now you have "HELP" pages to suit your changes and needs. SO easy!

No Doubt that all who have been using TW2 with the AERCO type of parallel interface have these pokes:

57999	127
58001	103
58008	127
58014	219
68015	127

Note that 127 is the port address.

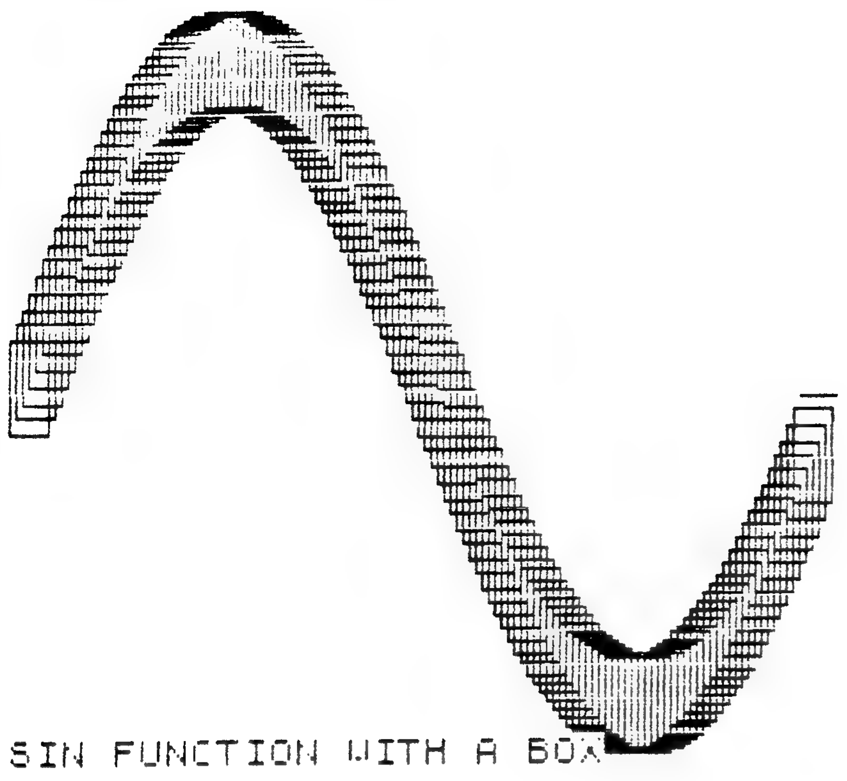
Special printer symbols with SYMBOL SHIFT in EXTENDED MODE:

Key	Symbol	Instead of
y	[	
u	]	
p	@	copyright
a	~	slanted quotes
s	|	solid bar
d	\	
f	{	
g	}	

Special printer symbols with SYMBOL SHIFT in the normal mode:

Key	Symbol	Instead of
h	^	up arrow
x	#	pound sign

Note: as this was written on MSCRIPT (my WP preference), 2 symbols were printed via embedded printer codes. The back slice is used for end of line or paragraph so could not be inserted in the text. The symbol for "d" is on the "s" key. This does not print in the text.



SEE PAGE 70 FOR THE PROGRAM

SALARY RAISE

Dick Wagner

In the Oregonian Newspaper Parade Magazine of June 7, 1992 there appeared in the "ASK MARILYN" column an explanation of a previously printed question. "You make $10,000 a year. You can have a $1000 raise at the end of each year, or you can have a $300 raise at the end of each six months. Which do you choose?"

This appeared to be a good study in short programming. I worked it out on the basis of using data statements first. Then came up with a formulas that would reproduce the data. Both methods are presented here. Using DATA requires 2 READ lines in order to display the results in column format. In this way one can selectively print the desired data. The READ a and READ b apply to the 6 month periods while READ c applies to the annual raise data. READ a is used to add the first 6 month salary to the second 6 month salary.

The key to this problem is understanding that "end" means a pay period has passed, like December 31 is the end of the annual pay period so the raise is given on January 1 (theoretically). Thus the raise for 1992 is given in 1993. Not exactly a play on words but probably a precise understanding in accounting practice.

```
   5 REM DICK WAGNER FOR THE PLO
TTER 6/10/92
  10 REM ASK MARILYN, OREGONIAN
PARADE MAGAZINE JUNE 7, 1992
  20 PRINT "YOU MAKE $10,000 A Y
EAR. YOU CANHAVE A $1000 RAISE A
T THE END OFEACH YEAR,";
  30 PRINT " OR YOU CAN HAVE A
  $300 RAISE AT THE END OF EACH
6 MONTHS."
  40 PRINT : PRINT "WHICH DO YOU
 CHOOSE? 1 OR 2? "
  45 PRINT : PRINT "1. $1000 AT
END OF                 EACH YEA
R                    2. $300 AT T
HE END                 OF EACH
1/2 YEAR"
  47 INPUT D: CLS
  50 REM RUN A TEST FOR 10 YEARS
, STARTING WITH 1992
  90 PRINT AT 0,0;"YEAR";AT 0,8;
"SALARY";AT 0,20;"SALARY";AT 1,6
;"$300 RAISE";AT 1,18;"$1000 RAI
SE";AT 2,7;"END EACH";AT 2,20;"E
ND EACH";AT 3,7;"6 MONTHS";AT 3,
20;"12 MONTHS"
  95 PRINT
 100 FOR n=0 TO 9
 110 LET s=10000
 120 READ a
 125 READ b
 130 READ c
 135 PRINT 1992+n;TAB 8;s+(a+b);
TAB 20;s+c
 140 NEXT n
 150 DATA 0,300,0,600,900,1000,1
200,1500,2000,1800,2100,3000,240
0,2700,4000,3000,3300,5000,3600,
3900,6000,4200,4500,7000,4800,51
00,8000,5400,5700,9000
 200 IF D=1 THEN PRINT : PRINT "
You selected a raise of $1000 at
the end of each year."
 210 IF D=1 THEN PRINT : PRINT "
So you lost on that one!"
 220 PRINT : IF D=2 THEN PRINT "
You selected a raise of $300 at
the end of each half year."
 230 PRINT : IF D=2 THEN PRINT "
You will win for a long time on
that choice."
```

INPUT CURSORS
Bill Dunlop

I like to get newsletters with bits of progtamming that I can use to dress-up my programming or to add to existing programs that I feel have "rough"spots in them. I have seen a variation of this one before but the Jan. '93 issue of UPDATE Mag. got me started playing around with this one again.

"On the TS2068 to get a question mark (?) displayed in an INPUT statement as a cursor you need to POKE 23617,236". It works, but, how much does that help in making a program clearer? Some, I agree, but if I am asking the user for a dollar amount why not make the cursor a flashing dollar sign ($) to keep money in mind during the INPUTing process?

We can do just that with a simple POKE of the keys!

```
10 POKE 23617,240: INPUT "Testing
"; a
```

Now enter this short program and make yourself some notes.

```
10 FOR x=255 TO 100 STEP -1
20 POKE 23617,x: INPUT "test";a$
30 REM press a key and ENTER
40 print x-1;" as cursor now "
50 LET a$=""
60 NEXT x
```

Some values just give a RAZZ but you can get past them with additional entries. Notice that the odd numbered POKEs change the cursor after the first character is input! Even numbers are stable until the ENTER key is pressed wherupon the system resets the proper code, thus if you want to use INPUT with the "custom" cursor you must re-POKE 23617.

A few of these even make sense as useful cursors! I like value 209 with its NEW changing to DATA when changing a value inside a program.

Changing > to a < for a "Drive #" prompt looks almost MesS-DOSian.

LARKEN LOCK!
Bill Dunlop

My youngest son loves games. Computer games. The ability of my 2068 to run both T/S or Spectrum games from disks has been a source of enjoyment, even when other tasks should be attended to. His mother has asked that I lock the computer up at times to help get his attention or as the result of some restriction.

As I often use the computer as a teaching tool for my boy even if he is on restriction how then to make the teaching programs useable while making the games unuseable? Being a programming nut the obvious solution of putting the games disks into a different location was not even considered.

He knows quite a bit about this system and has been known to load games when he is supposed to be studying.

How then to disable the game in such a way that if I wanted to use the games I can while still making them unavailable to him. Another goal was to leave the game itself unchanged, as reprogramming my over 140 games was more than I wanted to attempt. My 2068 should do that kind of repetitive tasks, not me.

The Larken system gave me as easy answer!

Thus was born a new program --- a LOCK & KEY for disk based programs.

See Page 86 for the program

ROUNDING NUMBERS

Dick F. Wagner

This handy program incorporates several features, tabulating numbers to a desired number of places to the right of the decimal point, and tabulating so the decimal point is in a fixed column to maked a nice looking display.

A few comments about the program--in place of the usual method of calculating the rounding (INT(v*100+.5)/100) for 2 places, lines 220 and 230 make the number of places variable so the user can input the desired number. Lines 330 and 350 display in the first column the number to be rounded while the second column displays the rounded numbers in justified form.

In operation the second column starts the TAB from a fixed point minus the length of the number, this makes it right justify. In order to use LEN N$, the calculated rounded number is converted to a string in line 300 so the length of the number can be used for Tab.

If a variable number of decimal places is required as an input, just change the GO TO in lines 330 and 350 to 200.

If a hard copy of the numbers is desired change the PRINTs in lines 330 and 350 to LPRINTs, or use COPY after the table is completed. Column headings can be added if desired.

This is a program useable on a 2068, TS 1000, or other BASIC language. If other than Sinclair some changes will probably be required.

```
  10 PRINT "    ****************
**"
  20 PRINT "    *ROUNDING NUMBER
S*"
  30 PRINT "    ****************
**"
  40 REM DICK WAGNER 7/93
 100 REM N=NUMBER TO BE ROUNDED
 110 REM DP=DIGITS TO THE RIGHTO
F DECIMAL POINT
 120 REM R=VALUE ROUNDED
 200 PRINT ;"HOW MANY PLACES TO
RIGHT OF     DECIMAL POINT FOR R
OUNDING? ";: INPUT DP: CLS
 205 PRINT ;"INPUT NUMBER TO BE
ROUNDED": PRINT
 210 INPUT N;
 220 LET A=N+5.5*10^-(DP+1)
 230 LET R=INT (A*10^DP)/10^DP
 300 LET N$=STR$ R
 310 FOR L=1 TO LEN N$
 320 IF N$(L)="." THEN GO TO 350
 325 NEXT L
 330 PRINT N;: PRINT TAB (20-LEN
 N$);N$: GO TO 210
 350 PRINT N;: PRINT TAB (21-L);
N$: GO TO 210
```

```
   5 REM A BRICK PROBLEM
  10 REM by Dick Wagner, Nov. 19
92
  20 PRINT "THIS QUESTION IS BAS
ED ON THE   PREMISE THAT ALL OF
THE BRICKS  INVOLVED WEIGH EQUAL
LY."
  25 LET W=4
  30 LET B=2*W
  40 LET C=B+W
  50 PRINT "IF A BRICK WEIGHS 4
POUNDS PLUS A HALF A BRICK, HOW
MUCH DOES A BRICK AND A HALF WEI
GH?";: INPUT C
  60 CLS : IF C<>B+W THEN PRINT
"WRONG, TRY AGAIN": GO TO 25
  65 PRINT : PRINT
  70 IF C=B+W THEN PRINT "GOOD T
HINKING, YOU ARE CORRECT"
```

USING DRAW WITH PLOT

Dick Wagner

An interesting display of PLOT with DRAW can be made with the following program. As noted in the REM statements, certain numbers make it possible to overlap vertical lines in the boxes. To easily show this, delete the last part of line 130 (the Y coordinate) and change it to 50. RUN and see that the lines overlap.

It should be possible to tack on any DRAW figure to a curve like this. The Y coordinates in line 130 will need to be adjusted to keep the figure within screen boundaries or the program will stop.

```
   1 REM
  10 REM PLOT A SIN CURVE WITH A
 BOX
  20 REM PRESS BREAK WHEN FINISH
 ED TO DELETE MESSAGE "B INTEGER
 OUT OF RANGE  1000:1"
  30 REM numbers .05 in LINE 130
 , 251 in LINE 140, and 12 in LIN
 E 900 are important for overlapp
 ing vertical lines
 120 PAPER 0: INK 7: BORDER 0
 130 FOR x=0 TO 2*3.14159 STEP .
 05
 135 IF x>6.1 THEN PRINT AT 21,0
 ;"SIN FUNCTION WITH A BOX"
 140 PLOT x/(2*3.14159)*251,SIN
 (x)*75+100
 145 GO SUB 900
 150 NEXT x
 160 GO TO 160
 900 LET A=12
1000 DRAW A,0
1010 DRAW 0,-22
1020 DRAW -A,0
1030 DRAW 0,22
2000 RETURN
```

SEE PAGE 67 FOR THE ILLUSTRATION

FEATHERS

```
   1 REM
   2 REM FEATHERS
   5 REM Converted from TRS to
 TS2068 by Ted Knyszek
  20 LET xp=0: LET yp=0
  30 LET s=PI/7.5: LET d=25
  40 FOR t=0 TO 1.089 STEP (s/20
 )
  50 LET d=d+3
  60 FOR i=t TO (t+PI) STEP s
  70 LET r=d*SIN (i)+.5
  80 LET x=r*SIN (i)+70
  90 LET y=r*COS (i)+86
 100 IF i=0 THEN GO TO 140
 110 LET xp=xp+.2: LET yp=yp+.2
 120 PLOT x,y
 130 DRAW (xp-x),(yp-y)
 140 NEXT i: NEXT t
```

CALCUATOR

Dick F. Wagner

Here is a little program that should come in handy for calculating values during program development. Put it in as a temporary subroutine either at the beginning or end. If at the beginning follow it with STOP. If at the end preceed it with STOP. This program still gives a flashing "L" or "C" indicating an input is required. For input, type the whole problem except the variable, as "5^11". The problem and answer is displayed on the lower screen area.

```
 10 PRINT #1; AT 0,0; INVERSE 1
 ;" What is your problem: Press
any key for INPUT "; PAUSE 0
 20 INPUT G$
 30 LET A=VAL G$
 40 PRINT #1; AT 0,0; INVERSE 1
 ;" The answer to your problem
is: ";G$;" ": PAUSE 0
```

YOUR SOCIAL SECURITY BENIFITS

by Dick F. Wagner

The following little program will provide helpful information concerning the date on which you will receive the optimum benifit from your Social Security payment. It gives the year and month at which you may retire, based on your birth year, so as to receive no early retirement penalty.

After RUN, make a printed copy of the screen in 2 steps with COPY.

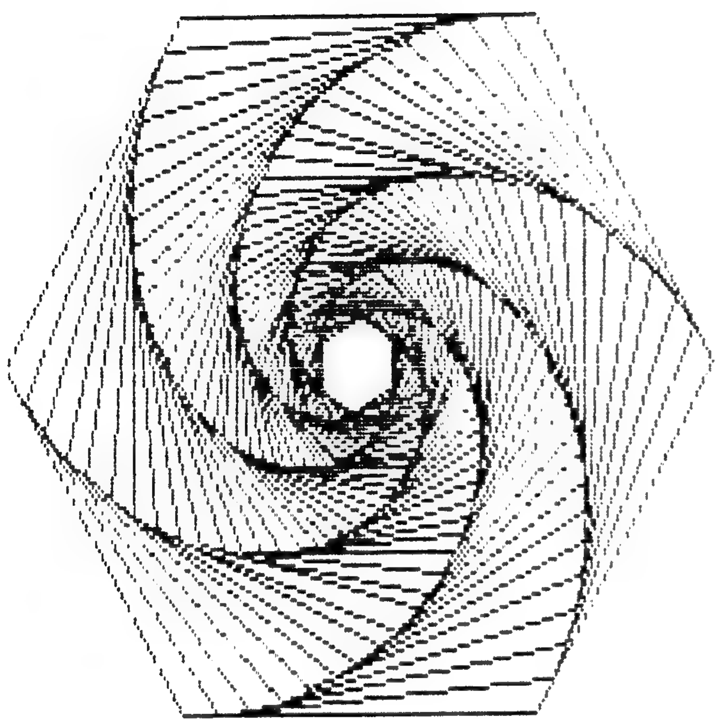
```
  10 PRINT "         SOCIAL SECU
 RITY"
  14 PRINT : PRINT "        by
 Dick Wagner"
  18 PRINT : PRINT "Information
 from Social SecurityAdministrati
 on": PRINT
  20 PRINT : PRINT "Contrary to
 the common concept  that at the
 age of 65 the ben-  ificiary can
   collect full socialsecurity be
 nifits upon retire- ment on that
  birthday, the full benifit pack
 age depends upon    one's birth
 date. The following table  displ
 ays the plan worked out by the S
 ocial Security Ad-  Ministration
 ."
  30 PAUSE 800
  40 CLS
  50 PRINT "Year of Birth";AT 0,
 19;"Full Benifit"
  60 PRINT AT 1,22;"at Age-"
  90 FOR n=1 TO 6
 100 READ a
 105 READ b$
 110 PRINT TAB 3,a;TAB 19;b$
 120 NEXT n
 130 PRINT "   1943-1954
 66 yrs"
 200 DATA 1937,"65 yrs",1938,"65
 yrs, 2 mo",1939,"65 yrs, 4 mo"
 210 DATA 1940,"65 yrs, 6 mo",19
 41,"65 yrs, 8 mo",1942,"65 yrs,
 10 mo"
 215 FOR n=1 TO 5
 220 READ a
 230 READ b$
 240 PRINT TAB 3;a;TAB 19;b$
 250 NEXT n
 300 DATA 1955,"66 yrs, 2 mo",19
 56,"66 yrs ,4 mo",1957,"66 yrs,
 6 mo"
 310 DATA 1958,"66 yrs, 8 mo",19
 59,"66 yrs, 10 mo"
 320 PRINT TAB 3;"1960 and after
  67 yrs"
 325 PRINT
 330 PRINT "No benifits before a
 ge 62, and  there is a permanent
  cut for    those who retire bef
 ore age 65, as much as 20%. CHEC
 K IT OUT!"
```

SPIRAL

By Dick F. Wagner

Here is a program that generates a nice spiral on the screen. To get a good symetrical screen dump use Stan Lemke's PRINTER program in the Jan, 1989 issue of UPDATE Magazine. It won't be exactly symetrical because the formulas do not produce an exact shape but it is close. The COPY command will gave a squashed image. By adjusting the formulas it may be possible to reduce the width by the amount the printed image is off.

```
  10 REM SPIRAL
  15 CLS
  20 LET c=COS (PI/3)
  30 LET s=SIN (PI/3)
  40 LET c1=COS (PI/36)
  50 LET s1=SIN (PI/36)
  60 LET sf=.95
  70 LET x=95
  80 LET y=0
  90 LET cx=130
 100 LET cy=88
 110 LET sc=1.16
 120 FOR j=1 TO 43
 130 FOR i=0 TO 6
 140 LET sx=x*sc+cx
 150 LET sy=cy+y
 160 IF i=0 THEN GO TO 190
 170 PLOT sx1,sy1
 180 DRAW (sx-sx1),(sy-sy1)
 190 LET sx1=sx: LET sy1=sy
 200 LET xn=x*c-y*s
 210 LET y=x*s+y*c
 220 LET x=xn
 230 NEXT i
 240 LET xn=sf*(x*c1-y*s1)
 250 LET y=sf*(x*s1+y*c1)
 260 LET x=xn
 270 NEXT j
 280 STOP
```



CALCULATING A SINKING FUND

by Dick F. Wagner

Sinking funds, an amount established or accumulated for a specific purpose, may be a part of a savings account or an investment plan. Consider calculating a part of monthly savings/investment to determine when a specific amount of accumulated interest can be withdrawn to meet a specific need at a specific time. How much must be dedicated to the account to generate the interest required?

This basic equation is the one to use: S=A(r)/((1+r)n). Where S is the amount of monthly payment into the account, A=amount accumulated (your need) r=interest in %/(12*100) n=number of periods as months.

This equation is reworked to fit the program needs.

```
 100 INPUT "How much do you need
 to accumulate? $";AMT
 110 INPUT "What is the annual
interest rate at %";R;" of the i
nvestment or saving account?"
 115 LET r=(R/(12*100))
 120 INPUT "How many months for
the investment to run to meet yo
ur need? ";n
 130 LET S=AMT*(r/((1+r)^n-1))
 140 PRINT "You will need to dep
osit $";S;" monthly for ";n;" mo
nths to produce $";AMT: GO TO 15
0
 150 INPUT "If the amount does
 accumulate to exactly cents the
n ENTER ""F"". If not then ENTER
 ""C""";x$
 160 IF x$="F" THEN STOP : IF x$
="C" THEN GO TO 170
 170 INPUT "If your deposits are
 not exactly the amount to the l
ast cent, use the next higher ce
nt and INPUT the new amount $";t
 200 REM A=t*(((1+r)^n)/r)
 210 REM recalculate the last pa
yment
 220 LET A=t*(((1+r)^(n-1)-1)/r)
 225 REM last payment LP=Amount
AMT-(A*(1+r))
 230 LET LP=AMT-(A*(1+r))
 240 PRINT "Yoy will be making
payments of $";t;" for ";(n-1);"
months, and a last payment of
$";LP;" to produce $";AMT;" in
interest"
```

MIND READER

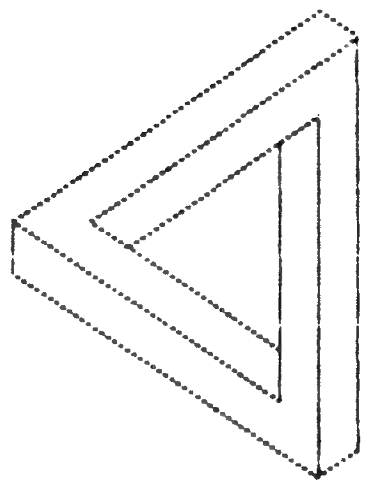
Bill Dunlop

The COURT JESTER is none other than new member Bill Dunlop, dba THE MAGIC CAVE. Maybe Bill has some more good programs up his sleave!

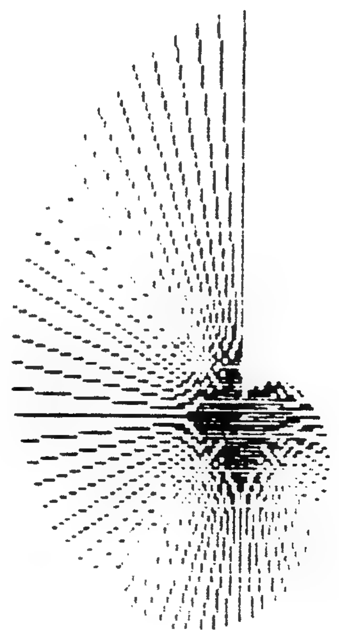
```
  5 REM **********************
 10 REM *ZX-PERT MIND READER *
 20 REM * BY THE COURT JESTER*
 25 REM **********************
 30 LET D=INT (PI)
 40 LET Q=D*D
```

```
  50 LET X=D-D
  60 PRINT "I WOULD LIKE FOR YOU
 TO WRITE   DOWN, ON PAPER, ANY
NUMBER OF   MORE THAN THREE DIGI
TS.  OK?"
  70 PRINT
  80 PRINT "PRESS ""ENTER"" TO G
O ON."
  90 PRINT
 100 PAUSE 4E4
 110 PRINT "NOW ADD THE DIGITS U
P ALONG THE NUMBER LIKE THIS. 12
3 WOULD BE  1+2+3 ETC. AND WRITE
 THIS NUMBER BELOW YOUR ORIGINAL
 NUMBER SO   THAT YOU CAN SUBTRA
CT."
 120 PRINT
 130 PRINT "IT SHOULD LOOK A BIT
 LIKE THIS",,"123  (1+2+3=6)",,"
-6",,"___"
 140 PRINT
 150 PRINT "PRESS ""ENTER"" TO G
O ON."
 160 PAUSE 4E4
 170 CLS
 180 PRINT "NOW AFTER YOU SUBTAC
T, YOU CAN  MULTIPLY THE NEW ANS
WER BY ANY  NUMBER ( NOT ZERO )
AND WE WILL USE THIS NUMBER."
 190 PRINT
 200 PRINT "IT SHOULD LOOK A LIT
TLE LIKE",,"123",,"- 6",,"---",,
"117",,"X 2",,"---",,"234"
 210 PRINT
 220 PRINT "PRESS ""ENTER"" TO G
O ON."
 230 PAUSE 4E4
 240 CLS
 250 PRINT "NOW I WOULD LIKE YOU
 TO CIRCLE  ANY SINGLE DIGIT IN
THIS LAST   NUMBER, BUT NOT A ZE
RO AS THAT  IS NOT REALLY A NUMB
ER.   OK??"
 260 PRINT
 270 PRINT "NOW GIVE ME ALL OF T
HE UNCIRCLEDDIGITS IN ANY MIXED
ORDER","THEN PRESS ""ENTER""AGAI
N."
 280 INPUT A$
 290 FOR Z=D/D TO LEN A$
 300 LET X=X+VAL A$(Z)
 310 IF X>Q OR X=Q THEN LET X=X-
Q
 320 NEXT Z
 330 PRINT ,,"NOW THINK YOUR CIR
CLED NUMBER   CLEARLY!"
 340 PAUSE 120
 350 CLS
 360 PRINT "YOU CIRCLED THE ";Q-
X,"AM I RIGHT ? Y OR N"
 370 INPUT R$
 380 PRINT
 390 IF R$(1)="N" THEN PRINT "I
SUGGEST THAT YOU LET ME DO    YO
UR MATH IN THE FUTURE AS YOU  MA
DE A MISTAKE IN YOUR MATH!"
 400 LET R$=""
 410 PRINT "DO YOU WANT TO TRY A
GAIN? Y OR N"
 420 INPUT R$
 430 IF R$(1)="N" THEN STOP
 450 GO TO 50
```

MY FAVORITE TRIANGLE
D.F.W.



SEE PAGE 37 FOR THE PROGRAM



SNAIL

THE CHECKWRITER

(or, How To Convert Decimal Digits to Printable Strings)
by Syd Wyncoop

I originally wrote the following program for a customer in Microsoft's Extended Disk Basic. It does not transfer to Timex Basic as well as I had hoped but will perhaps be of use to you.

I have often said that I much prefer the method Sinclair used for string handling over MicroSoft's, however, this time I found a reason to like MicroSoft Basic.

You will not that the strings are DIMensioned to the length of the longest required string of the group. MicroSoft Basic does not require the second parameter therefore the strings can be different lengths within the same array. I chose to eliminate some of the additional spaces this creates by use of the backspace character (CHR$ 8). This fix will not work on some printers and may require a loop to slice off the trailing spaces but, I will leave that to you.

The work is done in lines 110 to 250, with a companion subroutine at line 500. It would normally be called as a subroutine from within your program with the variable C$ containing a string of printable digits.

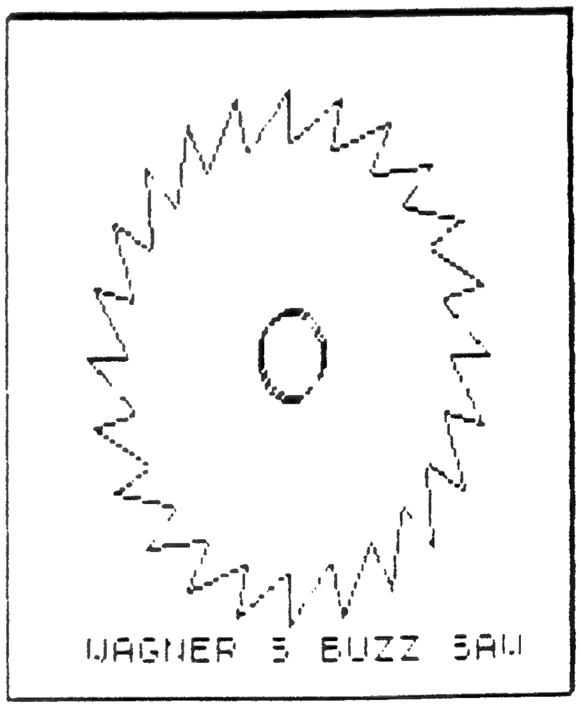
The first step is to ensure the number is padded out to two decimal places, which is why it is handled as a string. Then all leading 0's and spaces are removed. There is no error checking done for non-digits as the original program ensured the data was checked prior to placing it on the disk.

I converted it to Timex Basic as lots of individual lines in order to allow compatibility with the TS1000. 2068 users will be able to compact it greatly. (Editor's Note: Syd did not make it totally compatible; there are places where he put more than one command on a line. If you are going to use this on a 1000, you will have to add lines and do some renumbering).

Also, as it stands, the program will not properly write any amount that exceeds 9,999,999.99. You can of course change this.

The program will be of interest to you if you are running a business that writes a lot of checks. I have retained the copyright as it is part of another program, however, you are free to modify and use it for your own use.

**NOTE---
SEE PROGRAM ON PAGE 49**



SEE PAGE 26 FOR PROGRAM

INPUT!

Dick F. Wagner

After receiving Mr. Pegram's letter (last issue) on how to have more than one INPUT statement on one line, I did a bit of experimenting with his program. By the time I had finished I began to recall that I had a copy of an article on this subject. A search through my files of magazine articles brought up Robert Hartung's article on the subject in the CTM magazine, November 1985. CTM magazine has been laid to rest for some years now so the best way to give you his expertise is to copy that part of the article that pertains to INPUT.

"When I got my TS2068, I found even a simple thing like an INPUT command has some surprises. Of course I was pleased to find out that while the manual doesn't say anything about multiple-INPUT lines when it tells about using multiple- statement lines, any legitimate command preceded by a colon may follow another on the same line. Some statements, such as a REM or GO TO or IF-THEN, will not allow the processing of statements following so should be the last one on a line. The exception is if an IF-THEN condition is met that does not call for a branch to another line, ex. g:

```
IF X THEN INPUT A(n): LET C=C+1
```

"This is interpreted: if X is not equal to 0 then INPUT A(n) and increment C, and if X=0 then skip this INPUT and the increment of C and go on to the next line. This is very useful but requires that all statements following the IF-THEN on this line be dependent on its condi- tion(s) being met.

"Multiple-INPUTs with their respective prompts may also be used on a single line when separated by a semi-colon, comma, or apostrophe instead of a colon. This has the advantage of being able to see all of the inputs of a series at the time they are entered without cluttering up the working display on the screen. As in a PRINT line, temporary attributes of INK, PAPER, FLASH, as well as TAB n may be used for emphasis in a prompt if followed by a semicolon, comma, or apostrophe.

"Some PC's, such as the C-64, provide for an automatic numbering of inputs made in a FOR-NEXT loop. On the 2068, if we try something like FOR N=1 TO 5: INPUT n,A: NEXT N we find that both the n and the A are interpreted as requiring an INPUT. However, if we change this to FOR N=1 TO 5: INPUT STR$ n,A:NEXT n now we have as a prompt the current value of the loop variable n.

"But how about using a string in a prompt which is used for several different inputs? If we try INPUT A$,X this again is interpreted as calling for inputs for both A$ and X. What to do? While I was looking through a listing for the Spectrum in a ZX Computing magazine one day, I saw that the INPUT prompts contained both variable and string names enclosed in parentheses, so I tried this on my TS2068. Viola! it seems the Spectrum manual describes this as standard procedure but somehow it got lost in the trans- lation to the colonies by Timex.

"I have tried to put this in the following demo. It also includes a way of aborting a numeric input without changing the stored value (assuming that a letter o is not used as a variable name, which is not good practice anyway). The poke in line 70 provides automatic scrolling when the screen display is filled. Note also that when the ON ERR function is activated, at some point the program should pass through an ON ERR RESET to avoid creating a monster that eats up STOPs and BREAKs as errors and keeps right on going!

"When you begin adding peripherals to your computer, the INPUT command from the computer to the device, or vice versa, may also follow the formats described above, including multiple statements. A command such as INPUT # s,v will be read as an input of variable v into the I/O stream # s and will thus equate the device at this I/O port with the computer's own built-in hardware.

Robert D. Hartung

Addendum: A recent communication from Bob adds this additional bit of information:

"The above N loop example could also be written using INPUT (n),A instead of INPUT STR$ n,A. To input a string defintion without displaying the bracketing quotes, something like this may be used--INPUT LINE a$.

BAR CODE?

By: Bill Dunlop

If your business is small enough to be run using only common sence and your trusty 2068 you already know that the equipment to use IPC codes is way beyond your pocketbook. Fear not, help is near! Try Bills' Alphanumeric Resource.

Using a bit of thinking and our faithful 2068 we can "code" every item in our shop individually or by type and supplier. Using 5 symbols we can "code" over 14 1/2 MILLION items so there is little chance that any one-person type business will run out of space.

With only 5 symbols to read I do not need a laser or UPC bars to identify anything, plus, the symbols I use are all ones I recognize already, no new Greek or odd cyrillian alphabets here. The keyword is alphabets. Our computer already uses numbers as letters, see your manual for the ASCII tables, so why shouldn't we be able to use the letters as numbers. We can. If we use them in the same order that our computer uses them then any database program can order them and recall them easily. The labels could even be computer generated.

The code goes like this: 0 thru 9 is followed by the upper case alphabet (A thru Z) followed by the alphabet in lower case (a thru z) thus giving us a "number" set of 62 characters(10+52) for each set. Thus the "number" following 00009 is 0000A and the "number" following 0000Z would be 0000a. This goes on until the set is full and the next set starts, thus 0000z + 1 would be 00010. And on thru to zzzzz which in decimal is 14,776,397! And you thought converting from decimal to binary or hex was fun!

If you cannot forsee ever needing that large of a number you may be happier using 4 digits giving a maximum "number" of zzzz which is 238,389 decimal.

PS: you may want to censor out some combinations, as the State has learned with licence plates!

Bill's Alphanumeric Resource
```
 5 LET X=48
 10 LET Y=0
20 IF X>57 AND X<65 THEN  GO  TO
80
   REM non-alphanumeric
30 if X>90 AND X<97 THEN  GO  TO
80
   REM non-alphanumeric
40 IF X=123 THEN STOP
50 PRINT Y;
60 PRINT " = "; CHR$ X
70 LET Y=Y+1
80 LET X=X+1: GO TO 20
```

BOX THE TITLE

Dick F. Wagner

Last month we provided a program to generate a double line box. This method is suitable for a screen display but not practical for text to be printed on a large printer. This article deals with a solution for a print-out of text, such as I have here.

I am using MSCRIPT V5.3 on my 2068, and Epson LQ 570 printer. The method will be somewhat similar for Tasword II. Any printer that is Epson compatible with the same graphic symbols, and character codes for them should work, as long as MSCRIPT works with that printer.

The ASCII character codes used for making a double line box are as follows:

ASCII Assign Code	Character	Description	MSCRIPT Codes
201	╔	Top Lt cor	1
187	╗	Top Rt cor	2
200	╚	Bot Lt cor	3
188	╝	Bot Rt cor	4
205	═	Top & Bot	5
186	║	Ends	6

IBM compatible computers have these ASCII codes for graphic characters included in the character sets but they are not available in Sinclair Basic so the characters cannot be shown on the screen. They could be designed with the user graphic keys but that is a lot of work and they would not work for word processing. The printer codes would still have to be used.

MSCRIPT has a method of embedding printer codes in the text that uses the copyright symbol, ┐, plus a number from 0 thru 9. Note that the ┐ is not the same symbol used in MSCRIPT, which is a larger symbol. I have arbitrarily assigned the MSCRIPT codes as per the table. The screen will not display the printed box but I have used asterisk symbols to make a box of sorts. There will be as many asterisks as there will be ┐s along with the necessary printer and computer codes.

The first step is to assign the printer codes as prescribed for MSCRIPT by using $#1=201/1,#2=187/1, etc. Note that the $ is only for this text as the character is the right pointing arrow on the T key. If I used the correct symbol here the numbers would not print.

The second step is to generate the text and the box around it. It will look like this:

```
*****************
* BOX THE TITLE *
*****************
```

There are 17 asterisks across and 3 lines down. The box can easily be larger than this but not smaller for this text. Now we know how much space to allow for the box and printed text.

The third step is to replace each asterisk with the corresponding copyright symbol and MSCRIPT code. The top line is 1,(15)5s, and 2. The second line is 6, BOX THE TITLE, 6 and the last line is 3, (15)5s, and 4. Each number is preceeded by ┐. With MSCRIPT each time this combination is inserted in the text, the line moves right 2 columns to make room for the input. Just erase the asterisks with the backspace key.

The box and title will look like this, using the $ symbol for the copyright symbol:

```
$1$5$5$5$5$5$5$5$5$5$5$5$5$5$5$5$2

$6 BOX THE TITLE $6
$3$5$5$5$5$5$5$5$5$5$5$5$5$5$5$5$4
```

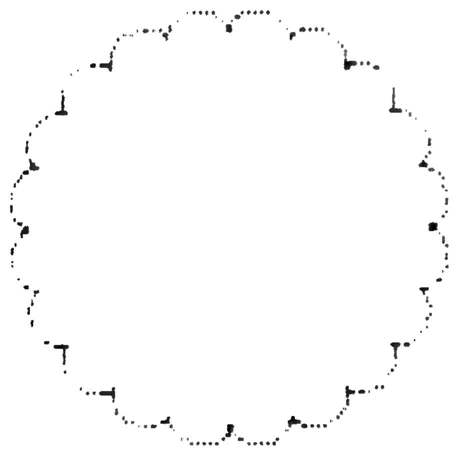
I have made this rather detailed but in normal use it is really quite simple, particularly in that you are using the copyright symbol.

Placement of the box is simple with MSCRIPT. Simply use the left margin command LM followed by the margin desired. This article is written with 64 columns with a left margin of 8. If the reader prints it out, the box will be at LM 26 to be centered. Remember to cancel these commands with LM and 8.

My columns for THE PLOTTER are printed with a left margin of 5 and line length of 36. The title box would then be printed at LM 5+(36-17)/2=14.

I'll try to accomplish this boxed title thing with the Tasword II word processor and give a report on it.

EPICYCLOID EXAMPLE



20:1

EPICYCLOIDS

Dick F. Wagner

An epicycloid is a type of curve that is formed when a smaller circle rolls around the ourside of a larger circle. The curve is the trace of a single dot located on the smaller circle. This can be illustrated a roller revolving around a stationary wheel, or a small gear revolving around a larger stationary gear.

Just visualize a painted gear tooth on the smaller gear. As the smaller gear revolves around the large gear, the painted tooth makes a path up and around, back to the stationary gear, but having progressed a fixed distance from the starting point each time the painted tooth contacts the fixed gear wheel.

The relationship of the radius of the larger gear and the smaller gear will determine the number of times the painted gear tooth makes contact with the larger gear in one complete roll around the big gear. If the gears are of the same radius then the rolling gear will make one complete revolution for the painted tooth to return to the starting point. If the larger gear has 2 times the radius of the smaller gear, the painted gear tooth will touch the large exactly twice, once at the midpoint of the circumference of the large gear, and when the painted tooth returns to the startng point.

The curve produced by the painted tooth is called an epicycloid. This is different from the curve produced by an inner gear or roller revolving inside of a larger circle, which is called a hypocycloid. The program that follows produces epicycloid curves.

The material for this article was developed from an article in the June 1984 issue of Creative

Computing, authored by Sheldon and Florence Gordon. Changes made to fit Sinclair Basic (2068) as the original program was for the TRS 80 Color computer with Extended basic.

To run the program the reader will need to decide on the ratios of the large circle and the small circle , in terms of the radii, such as 10:5, 8:2, 18:3, etc. Too large a radius will run off the screen . Numbers can be fractional or decimal, such as 10:2.5. Input your choices as called for, the large circle first.

Because of using the PLOT command in place of the original short line drawing method available in MS Basic (GWBASIC) the program produces dots in place of short lines. Thus the picture on the screen may be more difficult to interpret with complicated radii ratios and relative large steps.

Line 290 can be changed to smaller steps by changing ST to ST/2 which will take twice as long to plot the curve. Note that line 360 eliminates the display of the end of the run.

I like to know the origin of equations when I copy similar programs. The equations for the values of X and Y are as follows:

```
X=(A+B)*COS(T)-B*COS((A+B/B)*T)
Y=(A+B)*SIN(T)-B*SIN((A+B/B)*T)
```

where B is the radius of the outer rolling circle and A is the radius of the larger fixed circle. The larger circle does not visually appear on the screen but is the inner points or cusps of the path of the small circle.

This program can be converted back to the original by adding 4 lines:

```
245 SCREEN 1
270 LINE (H,V)-(H,V),PSET
340 LINE -(H,V),PSET
370 END
```

If you are using color then add it to the screen command.

```
  1 REM -- EPICYCLOIDS --
    -- by D. F. Wagner --
 90 FOR T=1 TO 1200: NEXT T
 100 CLS
 110 INPUT "What is the large ra
dius ";AA: PRINT
 120 INPUT "What is the small ra
dius ";BB: IF BB>=AA THEN GO TO
110
 130 LET C1=AA+BB
 140 LET C2=C1/BB
 150 DEF FN X(T)=C1*COS (T)-BB*C
OS (C2*T)
 160 DEF FN Y(T)=C1*SIN (T)-BB*S
IN (C2*T)
 170 LET A=0: LET B=6.28*BB
 180 LET N1=-C1-AA: LET N2=-N1
 190 LET M1=N1: LET M2=N2
 200 LET D=(N2-N1)/255: LET E=(M
2-M1)/191
 210 CLS
 220 LET NR=20*(AA+BB)
 230 IF NR>400 THEN LET NR=401
 240 REM -- DRAWS GRAPH --
 250 LET H=INT ((FN X(A)-N1)/D+.
5)
 260 LET V=191-INT ((FN Y(A)-M1)
/E+.5)
 280 LET ST=(B-A)/NR
 290 FOR T=A+ST TO B STEP ST
 300 LET X=FN X(T)
 310 LET H=INT ((X-N1)/D+.5)
 320 LET Y=FN Y(T)
 330 LET V=191-INT ((Y-M1)/E+.5)
 340 PLOT H,V
 350 NEXT T
 360 GO TO 360
```

PRINTING FROM ARRAYS

Dick Wagner

The Epson printer manual for the JX-80 printer has some good programs that are adaptable to Sinclair BASIC. One of interest is a method of sending to the printer data that has been stored in arrays. The following program calculates the plot of a small circle. The calculations are for one quarter segment of a circle. This is converted into a mirror replica horizontally, and this half circle is then converted into another mirror replica, making a full circle. This is then sent to the printer in rows of 7 dots. It produces the printout in 6 passes.

The program gives a running record of the 21 rows of pixels as it reads each row and stores the results in an array. It then shows the print head rows being printed. This is done on the screen as only the circle is printed on the printer.

Some line comments will better illustrate the program. Line 5 is for my Oliger printer driver. Use whatever is required to get ASCII output correctly to your printer.

Line 7 sets the left margin for my Epson work alike printer (Panasonic KX-P1080i). Not required if you want to print without a left margin. I use this to use other parts of a page when experimenting.

Line 10 sets "N" for 21. When using 7 pins in the print head, "N" must be in increments of 7. The array is dimensioned here.

Line 30 assigns the circle formula to "D". Circle formula? This formula calculates the distance between each pixel by taking the square root of the horizontal distance + the vertical distance. Remember the formula for the hypotenuse of a right triangle? The method used by this program is to determine each pixel position within the square encompassing the circle. Pizels that are to be black are called 1a and white are 0s.

Line 40 simply calls a pixel near 20 a "1", otherwise it is a "0".

Line 60 is the counter and prints out a running record on the screen.

Line 70 is the printer code for my printer for line spacing. It is using 7/72 which is the CHR$ 7 part.

Lines 80,90, and 100 change the order in which the array is read, toggling the order in which the array is read. First it is read upside down when Z=1, then right side up when Z=2.

Lines 110-160 qre as follows. Line 110 loads the array rows from beginning "B" to end "E" in sets of seven. Line 120 prints a screen record of the computer's progress. Line 130 enters Graphic Mode and reserves "N" columns for graphics, "N" being the width of the array. Line 150 accesses the subroutine that calculates the pin patterns for each column. Line 160 closes the loop for each pass "P" of the print head.

Line 170 can be deleted as it simply cleans out the printer buffer with the Reset.

Lines 170-220 convert the ones and zeros to pin firing sequences. The firing pattern is calculated for each column of 7 pins. It examines the array vertically, one cell at a time. When it encounters a "1", it adds the appropriate power of 2 to "F". The ABS (P+6*S-R) is the difference between the current row "R" and the last row in this pass of the print head (P*S-A), and finally, line 220 sends "F" to the printer as a graphic pin pattern.

WARNING:
This is a slow method of producing a print out of any shape of figure.

```
   5 REM  LET /p=0/G: POKE 23300,
60: POKE 23301,3
   7 LPRINT CHR$ 27;"1";CHR$ 20
  10 LET N=21: DIM A(N,N)
  20 FOR R=1 TO N: FOR C=1 TO N
  30 LET D=SQR (R^2+C^2)
  40 IF INT (D+.5)=20 THEN  LET A
(R,C)=1
  50 NEXT C
  60 PRINT "T MINUS ";N-R: NEXT R

  70 LPRINT CHR$ 27;"1";CHR$ 7;
  80 LET B=N: LET E=7: LET s=-1
  90 FOR z=1 TO 2
 100 IF Z=2 THEN  LET B=1: LET E=
N-6: LET s=1
 110 FOR P=B TO E STEP 7*S
 120 PRINT "LOADING ROWS ";P;" TO
 ";P+6*S
 130 LPRINT CHR$ 27;"*";CHR$ 5;CH
R$ (2*N);CHR$ 0;
 140 FOR C=N TO 1 STEP -1: GO SUB
 180: NEXT C
 150 FOR C=1 TO N: GO SUB 180: NE
XT C
 160  LPRINT : NEXT P: NEXT Z
 170 LPRINT CHR$ 27;"@"
 180 LET F=0: FOR R=P TO P+6*S ST
EP S
 190 IF A(R,C)=1 THEN  LET F=F+2^
ABS (P+6*S-R)
 200 NEXT R
 220 LPRINT CHR$ F;: RETURN
9999 SAVE "ARRAY"
```

APPLE PROBLEMS

Dick Wagner

This story problem is for the student in your family who is working up through math and has some experience with story problems. This is a "true-life" problem and not just "made-up". I put it into story form to make it more interesting.

It might take a little explaining about typing in fractions. The process can be considered as the addition of a fraction to a whole number-- 5 1/2 is actually 5+1/2. The computer doesn't recognize the number the way we use it.

```
   1 PRINT "This is a story prob
lem for     students who can wor
k simple    problems."
  10 PRINT : PRINT "DICK is fond
 of home made apple sauce and us
es some apples the  size of soft
 balls to make      apple sauce.
 They are called    CRISPEN appl
es."
  20 PRINT : PRINT "The Fir Poin
t produce stand thatsells CRISPE
N apples is only a  few miles fr
om his home."
  30 PRINT : PRINT "A big bin ho
lding these apples  has a sign s
tappled to its side that says ""
45 CENTS A POUND. 10 POUNDS AND
OVER 35 CENTS A POUND""."
  40 PRINT : PRINT "DICK picks o
ut 8 apples and     hands them t
o the clerk who     weighs them
and finds he has    7 3/4 pounds
"
  50 PRINT : PRINT "How much wil
l the apples cost?": INPUT C
  52 IF c<>3.49 THEN PRINT "Wron
g, try again.": GO TO 50
  55 IF C=3.49 THEN PRINT "Corre
ct,$3.49 for 7 3/4 pounds.": GO
TO 60
  60 PRINT : PRINT "The clerk sa
ys that he can add  more apples
for a few cents     more, if he
has 10 or more      pounds."
  70 PRINT : PRINT "How many pou
nds must be added   to 7 3/4 pou
nds to get them for $3.50 (a tot
al of 10 pounds)?   NOTE:a whole
 number plus a      fraction mus
t be like 3+5/8.": INPUT P
  72: IF P<>2+1/4 THEN PRINT "Wr
ong, try again.": GO TO 70
  75 IF P=2+1/4 THEN PRINT "Corr
ect, 2 1/4 pounds"
  80 PRINT : PRINT "If 8 apples
weigh 7 3/4 pounds, about how ma
ny more apples must he get out o
f the bin to make   at least 10
pounds?": INPUT A
```

```
  82 IF a<>3 THEN PRINT "Wrong,
try again": GO TO 80
  85 IF A=3 THEN PRINT "Correct,
 3 apples."
  90 PRINT : PRINT "DICK picks o
ut 3 more apples    and finds th
at he has enough    now to weigh
 10 1/2 pounds."
 100 PRINT : PRINT "How much wil
l the apples cost DICK?": INPUT
C
 110 IF C<>3.68 THEN GO TO 100
 120 IF C=3.68 THEN PRINT "Corre
ct, $3.68 for 10 1/2 pounds of a
pples."
```

NUMBERS, LARGE AND SMALL

Dick F. Wagner

Our computers thrive on numbers, but the languages we often work with have limits. In GWBASIC (Microsoft) the uper limit is reported to be +-17014120000000000000000000000000000000 which is 701412 with 32 zeros, or put another way (the computer way) +-1.701412*10*power of 38 (1.701412*10^38) or 1.701412E38. Sinclair BASIC seems to limit at about 1.7014059E38 where the number to the right of E is the exponent.

Single precission floating point numbers have limits based on the computer language. Floating point arithmetic is a system where the digits are kept separate, such as the mantissa and the decimal point.

Try this short program to display some of these limits

```
10 LET X=2^32-48
20 LET Y=2^32-47
30 PRINT X;"  ";Y
```

The display should be 4,294,967,200 which is printed by the computer as 4.2949672E9 and 4.2949673E9. If the computer could display the complete last number it would end with 300. Remember that E is the number of places to the right of the decimal point.

I recall a draftsman in my department who used to stew about rounding off of numbers by his calculator as he considered that it should give him the correct answer. It was worse when I made a calculation with my calculator and the answer may not be exactly the same as his. There has to be a limit with our computers. It is interesting to try to hit the limits as set by the computer system.

For the person who likes to experiment or test, try printing the result of 2.0110326^126 and then try increasing and decreasing the last 6 to squeeze the highest number out of the system. The limit of 7 digits after the decimal point restricts dividing the 6 into decimals such as 61.

BUY FACTORY DIRECT
Mitchell's Wholesalers

(555)123-4567

EXXOFF OIL CORPORATION

3215 Easy Street
P.O. Box 99999-EXX
Houstin, TX 13445

TO: ______________
____
____

USE YOUR 80 COLUMN PRINTER FROM BASIC

by Syd Wyncoop

We had quite a few inquiries at the last meeting on using a large printer from BASIC. I have given a solution, verbally, before but it seems there is still a great of confusion. Here is an example program and description to help pave your path to gaining the use you desire from your printer.

First, a word of caution. The example program assumes that you have an Epson or compatible printer. Each printer has its own set of codes to place it into certain modes of operation. You may have to search your printer manual for the correct codes but the principle will be the same as described here.

You can send two kinds of information to your printer, text or control codes. Most dot-matrix printers also have a bit-image graphics mode but we will not discuss that today.

Text is sent as ASCII character codes. You can find a table of these codes in many books. Only printable characters (letters & punctuation) can be sent in this manner. The used is:

```
LPRINT;"Hello there!"
```

Often, you will want to format the text in some special mode, such as expanded characters or italics. This is where we need the control codes.

Control codes are simply codes that tell the printer to shift gears. Most are sent after a special control code, called Escape. This is so the printer knows that you are sending it a control codes, instead of text. It is these codes that must be supplied by the manufacturer of the printer, as they have not been completely standardized.

The typical method of sending the control codes is by pressing the escape key and another key at the same time. Unfortunately, the Times does not have an Escape key.

The other method of sending these control codes must be used. It is:

```
LPRINT CHR$ 27; CHR$ 52
```

You LPRINT the character code for Escape, which is 27, then whatever character codes are needed follow it. The above example will turn on the italics mode for Epson printers.

You must refer to your printer manual for the necessary control codes for your printer.

There are some "standard" control codes. They will appear on the ASCII chart as the unprintable character codes, 0-31. These are sent without the Escape code.

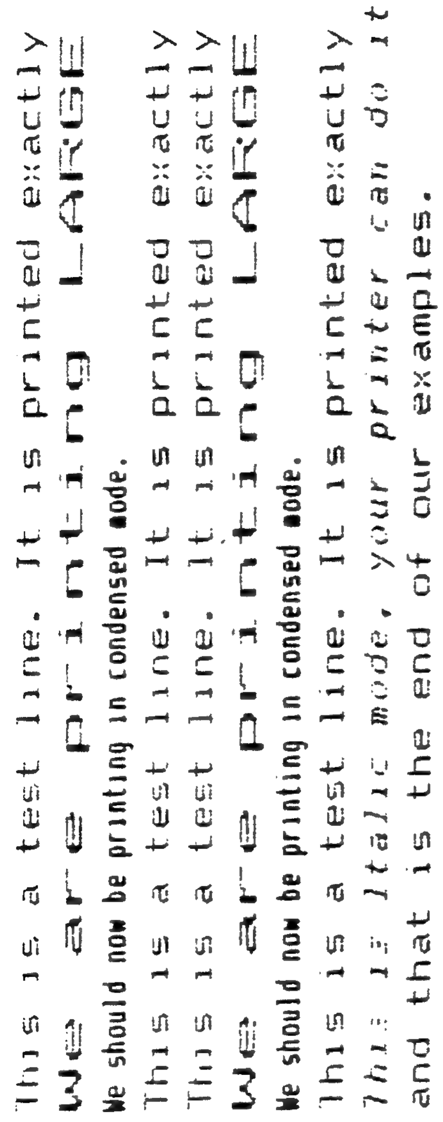
You must also be careful how you terminate your control code line. Just as in BASIC, anything other than a semi-colon will cause the printing to begin on the next line.

Experiment some. If you are still having difficulty, bring your printer manual to the next meeting. We will discuss the problem then.

```
   0>REM
       Cr w/LF.......Poke 64460
,10      Cr w/o LF.....Poke
6446
0,0        Width.........Poke
644
59,Width-1Print Mode.....Poke
64
456,1       Control Mode...Poke
6
4456,0       Turn On........Poke
26703,205                  &
Poke
 26704,251    Turn
Off.......Pok
e 26703,0                   &
Po
```

```
ke 26704,5
    1 CLEAR 64455: LOAD ""CODE
:
REM This line loads the printer
      driver code. You must have
        already saved a customize
d       version, as directed by
the       Aerco software that cam
e           with your interface.
    2 BORDER 0: PAPER 0: INK 9:
CLS:
                                RE
M Set screen attributes
  10 INPUT AT 0,0;"Please select
:"'"<P>rinter or <T>S 2040:"; L
INE a$:                         R
EM Select a printer
  20 IF a$="T" OR a$="t" THEN P
OKE 23703,0: POKE 26704,5: GO T
O 110:
REM Correct channel info to use
 the 80 column printer
  40 GO TO 10:               REM
 Trap Entry Errors
  50 INPUT AT 0,0;"Length of
lin
e to be printed?": LINE a$: REM
 Get width of printer  or
 line.
   60 IF VAL a$<1 OR VAL a$>150
T
HEN GO TO 50:
                    REM
 Trap entry errors
  70 POKE 64459,VAL a$-1:
    REM
SET LINE LENGTH TO PRINT
  80 INPUT "Do you need a line
f
eed after   the carriage return
s? (Y/N)"; LINE a$:
           R
EM This depends on how your pri
ntermicro-switches are set. Ans
wer Yif not sure.
  90 IF a$="N" OR a$="n" THEN P
OKE 64460,0: GO TO 300:        R
EM This sets up for no LF after
CR
 100 POKE 64460,1:           REM
This sets up for a LF after CR
 110 PRINT "OK, we should be
set
-up correct for the printer we
selected."'"You may re-select

printer optionby re-running the

program."'"You should now stu
dy the programlisting from line
 300 on to"'"determine how to u
se the printer driver to print
a ny text from "'"Basic.": LIST
30
0
 300 REM The following program
lines demonstrate the method yo
uneed to use to print to a larg
e printer, from Basic, with the
Aerco printer driver, whichis
supplied with their interfa
ce.
 310 LET Mode=64456:
      LET Print=1
        Let Cntrl=0:
REM We will use these variable
namesto make the listing clear
er.
 320 LPRINT "This is a test
line
. It is printed exactly as type
d."
 330 POKE mode,cntrl:          R
EM This will send the control c
ode for large letters to the pr
inter
 340 POKE mode,print:
      LPRINT "We are printing L
ARGE letters.":
REM We are back in text mode,
but weare printing large
letters
 350 POKE mode,cntrl:
      LPRINT CHS$ 15;:         R
EM This will send the control c
ode for small letters to the
printer
 360 POKE mode,print:
      LPRINT "We should now be
printing in condensed mode.":
REM We are back in text mode, b
ut weare printing small letters
 370 POKE mode,cntrl:
      LPRINT CHR$ 18;:
REM We have to turn off the shi
ft-inmode. This was not needed
for   shift-out as shift-out is
 only  active for one line of
text.
 380 POKE mode,print:
      LPRINT "This is a test li
ne. It is printed exactly as ty
ed.":
REM We are back in normal mode

 390 POKE mode,cntrl:
      LPRINT CHR$ 27;CHR$ 52: R
EM This will turn on Italic mod
e
```

```
 400 POKE mode,print:
     LPRINT "This is Italic mo
de, your printer can do it":
REM Print in Italics mode
 410 POKE mode,cntrl:
     LPRINT CHR$ 27;CHR$ 53:
REM This will turn off Italic m
ode
 420 LPRINT "snd that is the
end
 of our examples."
```



```
This is a test line. It is printed exactly
We are printing LARGE
We should now be printing in condensed mode.
This is a test line. It is printed exactly
This is a test line. It is printed exactly
We are printing LARGE
We should now be printing in condensed mode.
This is a test line. It is printed exactly
This is italic mode, your printer can do it
and that is the end of our examples.
```

NUMBERS
Dick Wagner

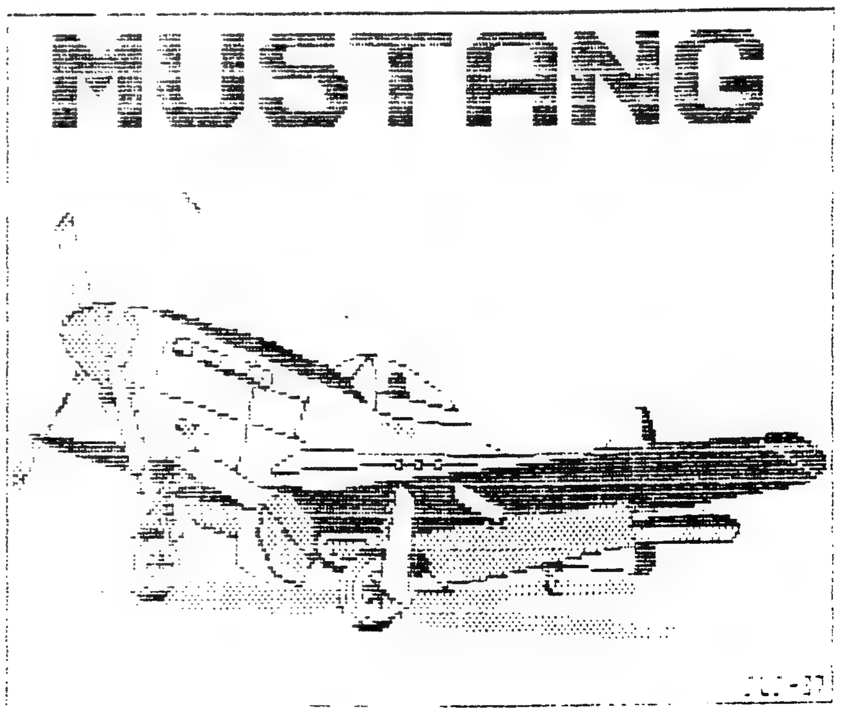
The user is to input an integer number between 1 and 100. The second input is the operator's calculation of N/3=A. The third input is the user's calcula- tion of A/5=B. The fourth input is calculation of B/7=C. Carry the calculation out to four or five places at least and round if desired.

The program uses the equation N=INT(70*C+21*C+15*C) to arrive at the number originally started with.

The inputs can be simplified by u sing N/3, A/5, and B/7. Once the last input is made the answer is displayed.

Several simplifications can be made in the process but that spoils the interest as the computer can display the correct answer immediately. Who is to say that it is not displaying the number first inputted?

```
  10 PRINT "SELECT AN INTEGER NU
MBER 1  TO  100. CALL IT N. MAKE
 THE FOLLOW-ING CALCULATIONS, CA
RRIED OUT AT LEAST 3 PLACES, AND
 I WILL TELL YOU THE NUMBER. THE
 CALCULATIONS ARE FOR A, B, AND
C ": PRINT "N=? ": INPUT N: PRIN
T N
  30 PRINT "N/3= A  ";: INPUT A:
 PRINT A
  40 PRINT "A/5= B  ";: INPUT B:
 PRINT B
  60 PRINT "B/7= C  ";: INPUT C:
 PRINT C
  80 LET N=70*C+21*C+15*C
  90 FOR Z=1 TO 5
 100 IF N>105 THEN LET N=N-105
 110 NEXT Z
 120 PRINT AT 12,6;"YOUR NUMBER
IS ";INT N
 125 PRINT
 130 PRINT "N=INT (70*C+21*C+15*
C)"
 140 PRINT "N=INT (70*";C;"+21*"
;C;"+15*";C;")"
```



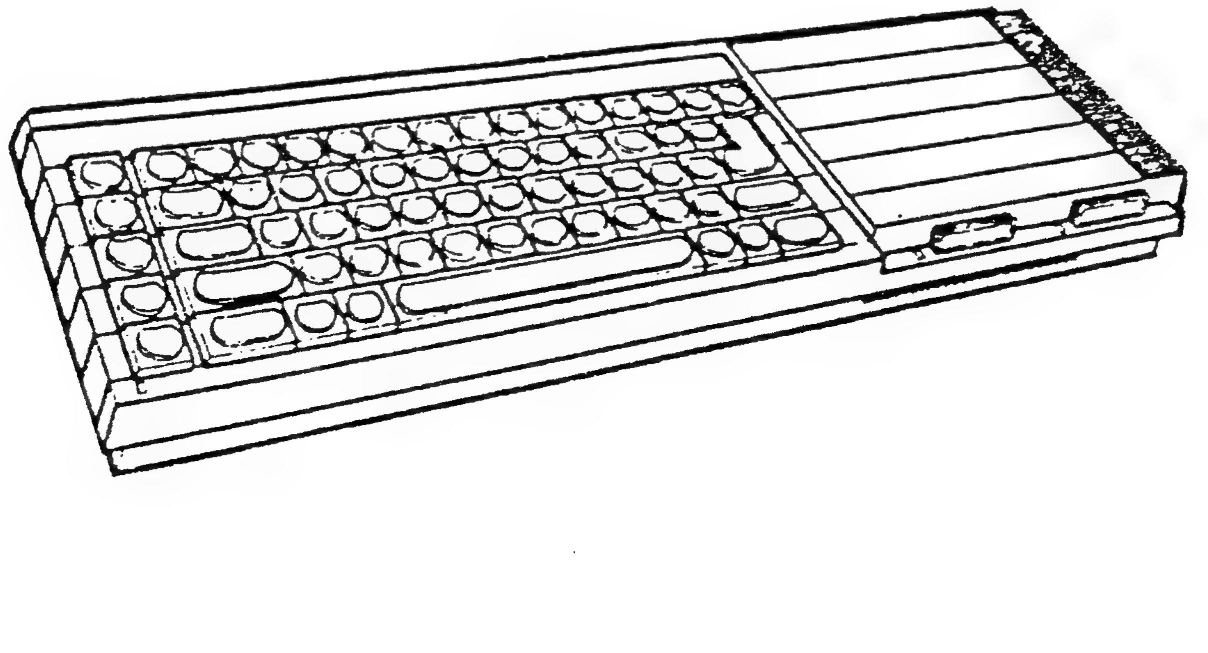
LARKEN LOCK!

Continued from page 68

```
  1 REM Larken Lock v1.2
  2 REM W.M. Dunlop, 1992
  5 INPUT "Key Code ";d
 10 IF d=dad THEN GO TO 20
 15 PRINT FLASH 1;" YOU BLEW IT
 KEITH! ": PAUSE 30: NEW
 20 INPUT "lock or unlock ";k$
 25 RANDOMIZE USR 100: GO TO 1
 30 RANDOMIZE USR 100: CAT  "",
 40 IF k$="l" THEN GO SUB 100
 45 IF k$="u" THEN GO SUB 200
 50 GO TO 20
100 REM lock
110 INPUT "lock name ";a$
115 IF a$="quit" THEN NEW
120 LET b$=a$: LET b$(LEN b$-2)
=CHR$ 0
130 RANDOMIZE USR 100: MOVE a$,
b$
190 RETURN
199 STOP
200 REM unlock
210 INPUT "unlock name ";a$
220 LET b$=a$: LET b$(LEN b$-2)
="."
221 LET a$(LEN a$-2)=CHR$ 0
230 RANDOMIZE USR 100: MOVE a$,
b$
290 RETURN
299 STOP
300 REM as a direct command
enter, LET d=your secret numbers
, before saving the program
```

Section 3:

SINCLAIR QL



THE QL QORNER (3/86)

by Vince Lyon

It may seem logical that the best place to begin a series of new columns on the QL would be a simple explanation of the basic differences between the QL and other Sinclair computers. Since, however, the QL is based on a new chip with new programming concepts and commands, the QL does not even look like a distant cousin of the 2068, nor appear to come from the same galaxy as the ZX81. There is, simply, no comparison.

A quick look at the program below clearly demonstrates this new, unique personality on the QL. As you will notice, the main loop of the program is only four lines long, and the majority of the remainder is a collection of defined procedures and functions that are called by name from the loop.

This ability to define any procedure and call it at will, by name is truly a Quantum Leap over 2068 BASIC. It is, to me, the single most exciting addition to the QL.

Although the program isn't meant to do anything fantastic, it does serve to demonstrate a few of the new commands that are available to the programmer. In the example, "FILLS" fills window #1 with periods which serve no purpose other than to prove that as one moves the cursor around, the periods are not erased.

OVER-1: This command reverses the Pixels at the print location. In this example, it permits the cursor to move about without overwriting screen data.

LINE: Allows a user to set up a variable for use ONLY within a DEF- ined procedure. If the same variable is used within the main program, it will not alter the variable in the DEFined procedure (and vice-versa).

THEN RETurn x: A nifty, fast way to say if x=1 then let y=1

Well, those are a few of the new commands used in this short sample program, and there are many more yet to explore (at least 25). These few examples, however, clearly help me to establish the concept that the QL is indeed unique, with little that appears related to the 2068. If one were to wish to convert this program to run on the 2068, it would be at least twice as long and much slower.

If you have a QL, enter and test the program. If you have a 2068, try to convert it.

The remainder of the program prints a cross-hair cursor on the screen which can be moved about by use of the cursor keys. The keys are read in "key-read", the position is updated in "position" and the cursor drawn in "lne" (since LINE is a QL command, it could not be used).

Let's look at a few of the new commands used in this program:

```
100 LET a=50:b=50:s=0
110 PAPER 0:=INK 7:BORDER 0,0:CLS
120 PRINT FILL$ (",",665)
130 lne:disp:wprint
140 REMark ------MARK LOOP-----
150 REPeat loop
160 Position
170 IF s=1 THEN EXIT loop
180 END REPeat loop
190 REMark -----END MAIN LOOP---
200 STOP
210 REMark --------------------
220 DEFine Procedure lne
230 REMark--------------------
240 OVER-1
250 LINE a,b TO a+10,b|LINE a+15,
    b TO a+25,b
260 LINE a+12,b-12 TO a+12,b-2:LI
    NE A+12,B+2 to A+12,B+12
270 OVER 0
280 END DEfine
290 REMark--------------------
300 DEFine FUNction keyread
310 REMark--------------------
320 LOCal q$
330 REPeat loop
```

```
340 LET q$=INKEY$(-1)
350 IF q$=CHR$(192) THEN RETurn 1
360 IF q$=CHR$(200) THEN RETurn 2
370 IF q$=CHR$(208) THEN RETurn 3
380 IF q$=CHR$(216) THEN RETurn 4
390 IF q$=CHR$(248) THEN RETurn 5
400 END REPeat loop
410 END DEFine
420 REMark--------------------
430 DEFine PROCedure Position
440 REMark--------------------
450 LOCal x
460 REPeat options
470 x=keyread
480 SELect ON x
490 =1:lne:LET a=a-1:lne:wprint
500 =2:lne:LET a=a+1:lne:wprint
510 =3:lne:LET b=b+1:lne:wprint
520 =4:lne:LET b=b-1:lne:wprint
530 =5:LET s=1:RETurn
540 END SELect
550 END REPeat options
560 END DEFine
570 REMark--------------------
580 DEFine PROCedure disp
590 REMark--------------------
600 WINDOW #2,110,10,200,175
640 DEFine PROCedure wprint
650 REMark--------------------
660 PRINT #2," ";a;" ";b
670 END DEFine
```

THE QL QORNER (4/86)

by Vince Lyon

Recently, I read a book which was supposed to be on superBASIC for the QL. However, than a little surprised when I came to the section on graphics. It seems that the author thought the graphics potential of the QL was best utilized by loading the included software and reading your manual on QL Easel.

While that may be well and good for those select few who don't wish to include charts or graphs within their own programs, it certainly is not for those who want to design real business programs.

It is not a great chore to use the graphics functions available from BASIC to create charts and graphs. And, the speed of the QL, makes it possible to use charts more quickly from within your own programs than it is to stop and load Easel.

Rather than explain in detail the entire below program (you should by now understand most of it), I'll let it stand on its own merit. You will find it easy to include in your own programs.

For those who may interested, I have another version of the program available (SASE) which adjusts the pixel width of the vertical bar chart, depending on the number of entries (up to 250).

If you have any QL programming hints, questions or applications, we would like to hear from you. Columns should reflect more than just one concept if they are to be a usable resource for us all.

```
10 MODE 4:WINDOW #1,450,165,35,0:
PAPER #1,0:INK #1,7:BORDER #1,1,7:
CSIZE #1,0,0: CLS
20 WINDOW #0,450,25,35,167: PAPER
#0,2: INK #0,7: BORDER #0,1,5:
CLS #0
30 high = 0: low = 100: ac= 0
40 UNDER 1: STRIP 3
50 FOR k = 0 to 19
60 LET lth = RND (2 TO 65)
70 ac = ac+lth
80 IF lth>high THEN LET high = lth
90 IF lth <low THEN LET low = lth
100 AT K,0: PRINT " "; TO lth;
"   ";lth
110 NEXT k
120 PRINT #0,,,,"Horizontal Bar
Chart"\,,"High = ";high;"   Low =
";
low, "Average = ";(ac/20)
130 FOR k = 10 TO  STEP 10
140 LINE k,0 TO k,175
150 NEXT k
160 PAUSE
170 UNDER 0: STRIP 0
180 CLS: CLS #0
190 av = 0
200 FOR k = 1 TO 55
210 LET hgt = RND (5 to 150)
220 av = av + hgt
230 BLOCK 5, hgt, k*8,160-hgt, 5
240 NEXT k
250 PRINT #0,,,, "Vertical Bar
Chart"
260 PRINT #0,,,, "Average = ";
```

```
(av/43)
270 PAUSE
280 CLS: CLS #0: POINT 0,0
290 AC = 0: CNT = 0
300 FOR k = 0 TO 170 STEP 2
310 LET np = RND (5 TO 100)
320 ac = ac + np:cnt = cnt +1
330 LINE TO k,np
340 NEXT k
350 INK 2: OVER -1
360 For k = 0 TO 170 step 5
370 LINE k,0 TO k,175
380 NEXT k
390 FOR k = p TO 100 STEP 5
400 LINE 0, k TO 430, k
410 NEXT k
420 INK 7: OVER 0
430 PRINT #0\,,,, "Line Graph"\,,
,"
    Average = ";ac/cnt
440 LINE 0, ac/cnt TO 430, ac/cnt
450 PAUSE
460 CLS: CLS #0
470 PRINT #\\,,,,"End Of Demo"
480 PAUSE
490 CLS #0
```

ARTICLE/PROGRAM...

THE QL QORNER (5/86)

by Vince Lyon

I'm going to assume that now you find that you're getting more and more excited about your new QL. You probably are wanting to write some special programs. Maybe even a space arcade/adventure. BUT, designing the screen is a real chore.

Suppose you wanted the screen to display everything the pilot would see: 1) a center video display which can face fore or aft on command, 2) an overhead console which reports on weapon status, 3) a small panel on the right center which shows aiming reports and target range, and 4) a below screen panel which reports on fuel, range from base, life support systems, etc.

If you have anything besides a QL, then you better get out your manual on MC. With the multi-tasking QL, however, this screen display is a real snap. On the QL one can set up all the windows in advance (we could use a DEFine PROCedure), and can print to any window and data or calculation result. Each window can be a color paper and ink, a different size with a different border and can print in characters sized strictly for it.

The program below shows some simple window designs with different colors and borders. I didn't use the full range of all the available character sizes because I am sure you will want to experiment with the defined windows.

As you may not, Line #5 sets up a window which is re-designed by Line #10. That is to make sure the screen is cleared of all previously defined windows.

The rest of the program is just a demonstration of printing windows of different sizes at different screen locations.

Remember that Window #0 is always the input window. When the program ends, the cursor will be in that window.

If you don't want to write the world's greatest space game, you will still find many useful applications for windows in business and game programs.

```
5 MODE 4:WINDOW #1,500,250,0,0:CLS
#1
10 WINDOW #1,440,150,30,0,:BORDER
#1:,1,7:CSIZE #1,2,0:PAPER #1,1,:
CLS
20 WINDOW #0,440,30,30,155:PAPER
KL#0,0:CSIZE #0,0,0,PAPER #0,5:INK
#0,0:BORDER #0,1,7:CLS #0
30 CLS #1:CLS #0
40 INPUT #0;"enter your name ";z$
50 PRINT "HELLO ";Z$
52 PAUSE 120
53 PRINT\\\\"GOODBY ";Z$
54 PAUSE 120
55 MODE 4:WINDOW #,500,250,0,0:CLS
#1
60 BORDER #1,0,0:CLS
70 WINDOW #1,440,100,30,0:PAPER
```

```
#1,2:BORDER #1,1,3:CSIZE #1,2,0:C
LS
80 WINDOW #0,440,100,30,105:PAPER
#0,0:INK#0,7:CSIZE #0,0,0
90 INPUT #0;"Enter your name*";z$
100 PRINT "HELLO ";z$
110 PAUSE 120
120 PRINT \\\"GOODBYE ";z$
130 PAUSE 120
135 MODE 4:WINDOW #1,500,250,0,0:
CLS
#1
140 WINDOW #1,200,100,30,0:BORDER
#1,1,7:CSIZE #1,2,0
150 WINDOW #2,230,100,240,0:PAPER
#2,0:BORDER #2,1,7:CLS #2
160 WINDOW #0,440,75,30,100:BORDER
#0,1,7:PAPER #0,2:INK #0,6:CSIZE
#0,0,0:CLS #0
170 INPUT #0\\"Enter your name ";
z$
175 PRINT #0\\\\\\\\\\
180 PRINT "well ";z$\\"meet my
windows"\\"this is #1."
185 PRINT #2;:"this is window #2"
190 PRINT #0;"and this is window
#0
(the input window)"
```

QL QORNER (6/86)

by Vince Lyon

One of the new commands available to the QL user is SELect. At first glance, it may not appear to be much of an addition but, as the program below demonstrates, there is more to the command than meets the eye.
First, remember that from within a SELect, it is possible to revalue any variable, call an defined procedure or function or any task you may want to assign. It can work from keyboard input, or the result of computations within a program (IF X=1 THEN).

In the sample, it simply changes the [string and calls a procedure (the same procedure for all), but it could just as easily call a separate procedure for each SELect, or two or more procedures. An inventive basic programmer can find a multitude of possibilities here for fast and easy procedure calls.

A short side note - you may notice that this months program contains no compound commands. It seems that the editor was driven almost to drink in attempting to type all the listings into a WP program so that they would fit the 36 column format, resulting in an occasional error or two. To make his job easier, and perhaps yours too, I am trying to keep all lines under 36 characters. Programs will be direct listed to the printer to eliminate any future errors. Any program that requires longer lines will be listed in condensed print so be ready to read some fine print in the future.

Again, if you have any comments or questions, we would like to hear from you. That's how we can tell if you like or read the QL Qorner...

```
10 MODE 4
12 WINDOW #1,450,145,35,0
14 PAPER #1,5
16 BORDER #1,1,2
18 CSIZE #1,2,0
20 WINDOW #0.450,40,35,150
22 CSIZE #0,1,0
24 BORDER #0,5,7
26 CLS:CLS #0
28 mainmenu
30 INPUT #0\\"   SELECT A NUMBER:
";
fi$
32 errtrap
34 fi=fi$
36 findit
38 STOP
40 DEFine PROCedure mainmenu
42 CLS:CLS #0
44 PRINT\\\\"  SELECT TEST"\\
46 RESTORE 54
48 FOR count=1 TO 6
50 READ m$:PRINT TO 5,m$\
52 NEXT count
54 DATA "1","2","3","4","5","6"
56 END DEFine
58 DEFine PROCedure findit
60 SELect ON fi
62 =1:p$="one":printit
64 =2:p$="two":printit
66 =3:p$="three":printit
68 =4:p$="four":printit
70 =5:p$="five":printit
72 =6:p$="six":printit
74 END DEFine
```

```
 76 DEFine PROCedure errtrap
 78 IF CODE (fi$)<49 THEN redo
 80 IF CODE (fi$)>54 THEN redo
 82 END DEFine
 84 DEFine PROCedure redo
 86 GO TO 28
 88 END DEFine
 90 DEFine PROCedure printit
 92 CLS:CLS #0
 94 PRINT\\\\ TO 4,"This is "&p$
 96 PRINT #0\\,,"PRESS ANY KEY ";
 98 PAUSE
 100 CLS:CLS #0:redo
```

QL TIP

by Michael E. Carver

When using the PSION suite of programs provided with the QL, strange things happen when you "Quit" these programs. SuperBASIC programs will not accept the INPUT command. The only way around this is to reset the computer. Well, that is not the "only" way. By making some changes to the "boot" program which loads these programs, this and other problems can be avoided.

All four of these programs close windows 1 & 2. As these are the default channels for most aspects of SuperBASIC, they are expected to be the first and second channels opened on the QL by QDOS, all kinds of things happen when they are closed and then reopened.

I like the ability to exit the PSION programs and continue to use features set up when I turn on my computer. Such as features from the Super Toolkit II along with RAM Disks on my Trump Card. Unfortunately, reseting the QL wipes out all files on RAM Disk and any other features "booted up".

The answer is simple, don't close those important default windows! The following listing will provide a sample "boot" for Quill. All lines which close and reopen windows 1 & 2 have been deleted from the boot and two procedures have been included to redefine the size of these two windows. These alterations can be done to the other three PSION programs.

Now whenever I "Quit" a PSION program, I can operate my QL in SuperBASIC or use Toolkit features.(especially the line editor,ED)
One more problem may arise when quitting the PSION programs--where's the memory? Sometimes the PSION programs will grab some memory and keep it "locked up" after returning to SuperBASIC. I have been able to reclaim this memory by using the DEL_DEFB command from the Super Toolkit.

In the example below Quill can be re-entered by entering the keyword <quill>.

```
    1 GO TO 30000
   10 DEFine PROCedure quill
   40 CLEAR
   50 WINDOW 512,256,0,0:CSIZE 2,
1:CLS
   60 AT 2,11:PRINT "LOADING QL-WP
"
   70 AT 4,13:PRINT "version ";2.1
   80 AT 6,6:PRINT "copyright 1984
PSION LTD"
   90 AT 8,12:PRINT "word processo
r"
  100 WINDOW #0,400,20,35,215
  110 EXEC_W mdv1-QLWP
  120 mon
  130 END DEFine quill
30000 PRINT "input correct date
and time by"/'["sdate yyyy,mm,dd
,hh,mm,ss]'
30010 quill
31900 DEFine PROCedure tv
31902  MODE 8:WINDOW 512,256,0,0
31904  PAPER 0: CLS
31906  WINDOW 512,206,0,0
31908  WINDOW #2,512,206,0,0
31910  WINDOW #0,512,50,0,206
31912  PAPER 2: PAPER #2,1: PAPE
R #0,0
31914  INK 7: INK #2,7: INK #0,7
31916  CLS: CLS #0
31918 END DEFine tv
31920 DEFine PROCedure mon
31922  MODE 4: WINDOW 512,256,0,
0
31924  PAPER 0: CLS
31926  WINDOW 256,206,256,0
31928  WINDOW #2,256,206,0,0
```

```
31930  WINDOW #0,512,50,0,206
31932  PAPER 2: PAPER #2,6: PAPE
R #0,0
31934  INK 6: INK #2,2: INK #0,4
31936  BORDER 1,255: BORDER #2,1
,255
31938  CLS: CLS #2: CLS #0
31940 END DEFine mon
31924  PAPER 0: CLS
```

COMBINATION

by: James Edwards

```
    10 MODE 4:WINDOW 512,256,0,0:
PAPER 0:INK 7:CLS
    15 WINDOW 420,200,40,26:PAPER
0:INK 7:BORDER 1,7:CLS
    20 CSIZE 2,1:PRINT TO 12;
"COMBINATION":CSIZE 1,0
    25 PRINT "Combination was writ
tten for the 2068 by Vladimir and
Aleksandar Bulovic. With some mino
r changes, it now works on the QL.
Plus, you can play against thecomp
uter. Note: The computer is not th
at good but will give you good pra
ctice.
    30 PRINT "The playing board co
nsists of 64 boxes. The goal of th
e game is to get four boxes in a l
ine. You can win in any direction.
, diagonal, horizontal or vertical
  ."
    40 PRINT "To choose the column
you wish play in just PRESS key 1
to 8."
    50 PRINT\\:CSIZE 2,1:PRINT TO
7;"PRESS ANY KEY TO PLAY"
    60 PAUSE
    80 MODE 8:WINDOW 512,256,0,0:
PAPER 0:INK 7:CLS
    85 GO TO 200
    90 DEFine PROCedure BOARD
    95 title
   100 INK 7:FOR a=46.9 TO 115
STEP 7.85
   110 LINE a,26 TO a,89
   130 END FOR a
   140 FOR a=26 TO 95 STEP 7.85
   150 LINE 47,a TO 109.6,a
   160 END FOR a
   170 END DEFine
   200 name
   210 b=INT(RND(1 TO 2)):b=b*2
:d1=1:d2=8
   300 PAPER 0:INK 7:CLS:BOARD
   310 DIM s(34,34)
   350 CURSOR 200,200,:INK 7:PRINT
"1 2 3 4 5 6 7 8"
   360 DIM c(8)
   370 FOR x=1 TO 8:C(x)=17
   380 IF b=6 THEN b=2:END IF
   390 FOR f=1 TO 9
   400 PAPER b:INK 0:AT f+8,0:
PRINT"          "
   410 END FOR f
   430 AT 11,5-LEN (a$(b/2))/2:
PRINT ;a$(b/2)
 450 CSIZE 2,0:AT 13,0:PRINT TO 6;"
  IS":CSIZE 3,0
 460 AT 15,0:PRINT" PLAYING"
 480 PAPER 0
 500 FOR x= 1 TO 8
 510 IF c(x)<=1 THEN
 520 END FOR x
 530 GO TO 2000
 540 ELSE:END IF
 545 IF cpu=1 AND b=4 THEN
 547 d= RND(d1,d2):PAUSE 30:GO TO
570
 548 ELSE:END IF
 550 ch=CODE(INKEY$(-1))
 560 IF ch>48 AND ch<57 THEN
 565 d=ch-48
 566 ELSE
 567 GO TO 545:END IF
 570 IF c(d)>2 THEN
 575 g=25-c(d)
 590 INK b:FILL 1:CIRCLE (d*2+10)*
7.8 2/2+4,g*7.83/2-1.6,3.5:FILL 0
 595 x1=c(d):y1=d*2+10:s(x1,y1)=b
 600 GO TO 1000
 610 ELSE
 620 GO TO 545
 650 END IF
1000 e=48+b
1010 IF cpu=1 AND b=2 THEN
1020 d1=d-1:IF d=1 THEN d1=1:END
1030 d2=d+1:IF d=8 THEN d2=8:END
1040 ELSE :END IF
1190 b=b+2
1200 RESTORE 1210
1210 DATA 0,1,1,0,1,1,-1,1
1220 FOR z=1 TO 4
1230 READ o1,o2
1240 FOR y=0 TO 3
1250 FOR x=0 TO 3
1260 n=x*-2+2*y:IF n=0 THEN
1270 NEXT x
1280 GO TO 1500
1290 ELSE :END IF
1300 x1=c(d)+n*o1:y1=d*2+10+n*o2
1305 IF x1<1 OR y1<1 THEN
1310 GO TO 1340
1312 ELSE :END IF
1315 IF s(x1,y1)=b-2 THEN
1320 NEXT x
```

```
1330 GO TO 1500
1335 ELSE :END IF
1340 NEXT y
```

QL QORNER (9/86)

by Vince Lyon

Before we begin this month, I have heard from some that the programs thus far printed did not appear on the screen as intended. Since most of the QL's out there are running on TVs and not monitors, all the programs have been designed for a TV display. Before you enter this program, make sure that you are in TV mode. You can always redesign the program if you wish.

This month's program is just a fun little demo of the turtle graphics. As anyone who has experimented with turtle graphics can attest, this isn't a true example of the full capabilities of that turtle. But, it does serve to demonstrate the basic commands available on the QL.

Since, by now, you all know how to use DEFined PROCedures, I put all the turtle commands in a PROCedure called "doit". All the print locations & ink colors are contained within the DATA statements.

In using turtle graphics as designed for the QL, it is nice to note that the commands POINT & TURNTO may be variables, calculated from within the program or assigned by a DATA & READ statement.

Back to old business - We're still looking for additional input for this column. If you have discovered some amazing (or not) little trick, or have a question about a command, or a comment (civil), we would like to hear from you. Remember that this column is intended to be a resource for ALL QL users, and not just a place for me to present my views.

```
10 PAPER 0:CLS:CLS #0
20 FOR m=14 TO 18 STEP 2
30 RESTORE : PAUSE 60: CLS
40 FOR x=1 TO 5
50 READ a,b,c
60 doit
70 DATA 30,27,1,80,27,2,130,27,3
80 DATA 55,72,4,105,72,5
90 NEXT x
100 NEXT m
110 a=80:b=54:c=7:m=30: doit
120 FOR z=1 TO 7:c=z:doit:NEXT z
130 RECOL 0,2,0,2,0,0,0,6
140 PRINT #0\\,"That's all folks ";
150 PAUSE 240: RUN
160 DEFine PROCedure doit
170 POINT a,b
180 INK c
190 PENDOWN
200 TURNTO 0
210 FOR k=1 TO 28
220 MOVE m:TURN 90
230 MOVE m:TURN 90
240 MOVE m:TURN 90
250 MOVE m:TURNTO m*k
260 NEXT k
270 END DEFine
```

QL QORNER (10/86)

by Vince Lyon

To run this month's demo, have your QL in TV mode and a formatted cartridge in MDV1_.

Now we have a chance to look at a few of the facilities and commands we haven't explored before. Note line 35 (a%=k) which is the new QL method of assigning the INT command. When the program is run, you will notice that the QL always rounds properly without adding the .5 to the function as was necessary with the 2068.

The program than does some demos of the PAN and SCROLL commands, and a screen save and recall. If you run the program a second time, make sure to DELETE MDV1_display, since the QL will not automatically overwrite an existing file.
Now to some more serious business - I have been informed by my wife that my eyes are starting to look like two cherries in a glass of

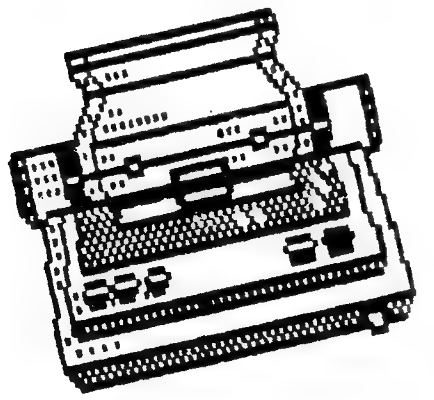
buttermilk. It seems that she thinks I spend too much time staring at a CRT trying to come up with little tidbits for the QL column. Since the only time available to me to write these ideas is usually after midnight, I guess she's probably right.

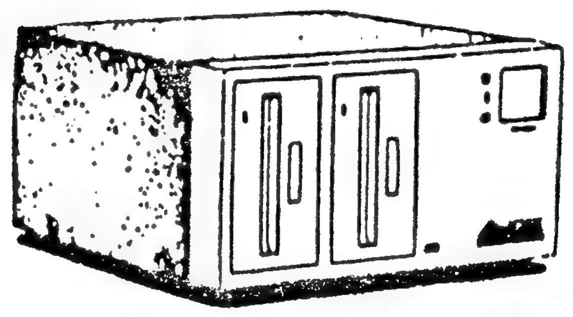
If this column is to survive, and my eyes return to normal, it's essential that there be more input. Several past appeals have not generated any responses, and I am about out of ideas to pursue. Your questions and comments, however, can give me a starting point for a column. So, if you want the QL QORNER to continue - write us about something - anything.

```
10 DEFine PROCedure show_int
15 PAPER 0:INK 7:BORDER 6,2,7
20 CLS:CLS #0
25 PRINT:PRINT
30 FOR k=1 TO 2 STEP .1
35 a%=k
40 PRINT TO 3;"k=";k;
45 IN a%+2
50 PRINT TO 2+(a%*10); "INT = ";a%
55 INK 7
60 END FOR k
65 PAUSE 240
70 END DEFine
75 DEFine PROCedure move_up
80 FOR k=1 TO 105
85 SCROLL -1
90 NEXT k
95 CLS
100 END DEFine
110 DEFine PROCedure move_down
115 FOR k=1 TO 185
120 SCROLL 1
125 END FOR k
130 CLS
135 END DEFine
140 DEFine PROCedure move_right
145 FOR k=1 TO 185
150 PAN 3
```

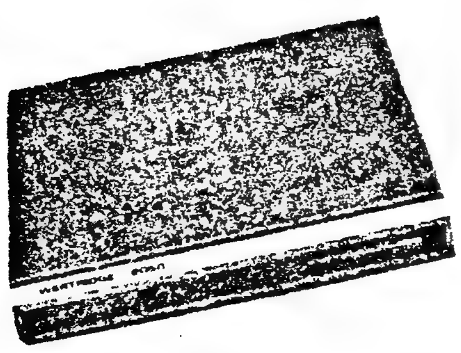
Section 4:

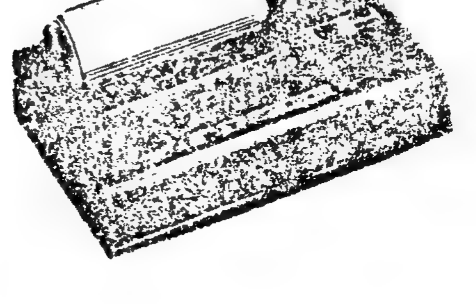
GENERAL INFORMATION





DISK DRIVES





The program name must be 7 characters or less. One problem occurred using this system. In Tasword II if you save your text, the program asks you for the name of your text so that the text can have a name to be stored under. The name is input to a string variable:

```
9000 SAVE A$ CODE m,n
```

To save on wafer, this would be:

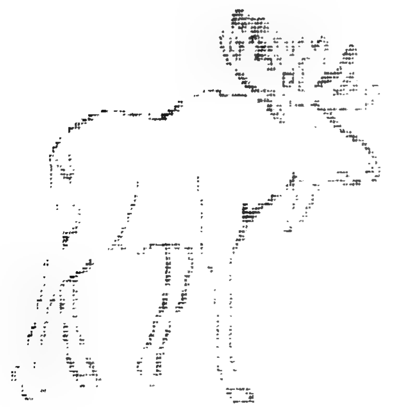
```
9000 SAVE "@#,A$"CODE m,n
```

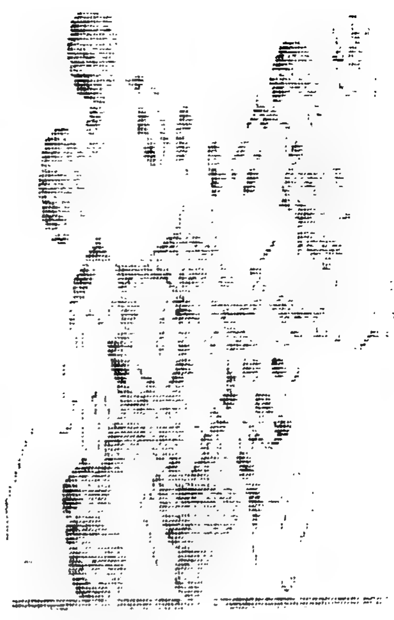
...which doesn't work. So, you must break Tasword II at this point and type SAVE "@#,program name"CODE m,n to save the text on wafer. (*)

The wafers themselves are marvels. They are small and compact and have a sliding dust cover to protect them that must be closed in order to be able to insert them into the drive.

I do not know enough about disk drives to compare the micro drive to them, but from what I have heard, the microdrive is slower than some and faster than others. All in all, if you have had tape recorder problems or you just wish for something better, then this is one way you should consider.

(*)Editor's Note: This is one way, there are others which involve going into the BASIC portion of TW II and changing the loading command.





GETTING MORE FROM YOUR 2050 MODEM

by Miles G. Rose

Here are a few ideas to try with your 2068 and 2050 modem. Through trial and experimentation, I found the port addresses and some of the modem calls, so I learned how to program some tricks of My own, in BASIC.

The modem seems to use only ports 115 and 119. port 119 serves to control the modem, while 115 carries the actual data. Here are the modem calls I use. I got these from the TIMEX database on CompuServe.

OUT 119,0	Hangs up phone
OUT 119,1	Hangs up phone. but doesn't stop carrier
OUT 119,31	Initializes phone for AUTO-DIAL
OUT 119,34	Starts Carrier tone if IN 119=133, the modem is connected

Port address 115 carries the data itself, in the form of ASCII code, which pretty well matches the character codes the 2068 uses. That being the case, it should be easy to whip up a BASIC program to handle data transfer. The routines I present here work once the modem connection has been made. Here's how to do that with SMART II.

Load and run SMART II. Once you are in the main menu, make sure that all the system parameters are established. I haven't gotten to that yet, so I do that from within the program. Make the call, and once the connection is made, and the info you need is coming, get to the main menu and EXIT to BASIC. Then you can RUN the following routines.

This will transmit data. A is the address where the character code is stored. In this case, 33280 is where Tasword stores code.

```
  10 LET A=33280
  20 ON ERR GOTO 60
  30 PRINT CHR$ PEEK A::OUT 115,
 PEEK A
  40 PAUSE 1.5
  50 LET A=A+1
  60 GOTO 20
  70 ON ERR RESET: LET A=A+1
  80 GOT 20
```

This routine will download data to address A, here arbitrary. The length of PAUSE here is critical. You must experiment to match it to the rate of the host.

```
  10 LET A=30000
  20 POKE A, IN 115
  30 ON ERR GOTO 80
  40 PRINT CHR$ IN 115
  50 PAUSE 1.5
  60 LET A=A+1
  70 GOTO 20
  80 ON ERR RESET: LET A=A+1
  90 GOTO 20
```

There should be a lot of things to do with these routines. They could be used in some sort of integration software, to create your own BBS or maybe even a terminal program in BASIC. Good luck.

EXTENDED CABLE PROBLEM

by Dick Wagner

Extended cables on the 2040 printer may run into problems when used with a TS 1000 computer equipped with a large keyboard. The first symptoms of a problem showed up when I connected my extended cable printer to the 1000/ Cricket House Keyboard. All keys worked fine except the shifted S, D, F and G. There was no action at all on these keys.

My first idea was to reduce the signal coupling between wires in the round shielded 3' cable. A new 13 wire flat cable was installed with a ground wire between each signal wire. This was an improvement, bringing all keys to life except the Shifted S.

The next step was to try a different KB resistor for KB0. Cricket House supplies a set of 5 resistors of about 56K OHMS to parallel the regulator 10K KB resistors. I found that a 1.5K resistor at KB0, which is the Shift key resistor, did the trick. Now I am able to use the large keyboard connected to the 1000 computer with an 18" cable and the 2040 with a 3' cable.

LOAD/SAVE METER

by M.J. Raymond

Finding some time on my hands this summer, I decided to do something about the load problems on my T/S 1000.

Looking through the Radio Shack semiconductor guide, I found a circuit for an easy VU Meter. With only a few changes and an hour or two with a soldering iron, this could be a useful addition to your computer. Construction of the circuit is easy and straightforward. I changed the resistors to variable ones to provide for easy adjustment. The circuit in the book is for 0 to 10 volts which is a little high for our needs. The power leads connect directly to the 9 volt input to the computer. The signal input is soldered to the tape inputs and that is all there is to it!

After connecting up the leads, it will help to do a little experimenting. First, adjust R1 so that the loudest volume lights the last light on the bar graph. This can be done without using the LOAD on the computer, but it has to be turned on. R2 controls the brightness of the display. Set this where you like. Now, close up the case. Try LOADing a program several times to find the max. and min. volume that the computer will LOAD at. Mark these limits on the bar graph and as long as you maintain the volume between the marks, you should have a good LOAD.

I have found this circuit not only useful in LOADing, but it also helped to align the heads on my two recorders.

PARTS LIST:

1-RS276-150	Circuit board	$ .79
1-RS276-1991	20 pin socket	.59
1-RS276-1992	18 pin socket	.49
1-RS276-081	10 section LED Bar Graph	3.00
1-RS276-1709	LM3916N IC	2.99
1-	5K Variable Resistor	.59
1-	10K Variable Resistor	.59
10 in.-	Light Wire from work bench	
2 in.-	Solder from work bench	
	TOTAL:	$10.00

All parts listed are available from your local Radio Shack.

(See diagram Page 108)

****

```
  5  REM Wafer Menu Driver
 10 REM 1985 RMG SOFTWARE
 15 REM main menu
 20 BEEP VAL ".4",VAL "10": POKE
VAL "23658",VAL "8"
 25 CLS : PRINT  INVERSE VAL "1"
;"*********************************
*"
 30 PRINT  INVERSE VAL "1";"****
"; INVERSE VAL "0";" WAFER INDEX
 DIRECTORY "; INVERSE VAL "1";"**
**";AT VAL "2",VAL "0";"*********
***********************"
 32 REM TYPE IN THE NAMES OR
      CODES FOR EACH OF UP
     TO 45 WAFERS, WHEN
     FINISHED, GOTO 20
 35 PRINT AT VAL "4",VAL "0";" 1
)records 16)  ppp   31)  555  "
 40 PRINT " 2)recipes 17)  qqq
32)  666  "
 45 PRINT " 3)books   18)  rrr
33)  777  "
 50 PRINT " 4)tapes   19)  sss
34)  888  "
 55 PRINT " 5)games   20)  ttt
35)  999  "
 60 PRINT " 6)videos  21)  uuu
36)  AAA  "
 65 PRINT " 7)    ggg  22)  vvv
37)  BBB  "
 70 PRINT " 8)    hhh  23)  www
38)  CCC  "
 75 PRINT " 9)    iii  24)  xxx
39)  DDD  "
 80 PRINT "10)    jjj  25)  yyy
40)  EEE  "
 85 PRINT "11)    kkk  26)  zzz
41)  FFF  "
 90 PRINT "12)    lll  27)  111
42)  GGG  "
 95 PRINT "13)    mmm  28)  222
43)  HHH  "
 100 PRINT "14)   nnn  29)  333
44)  III  "
 105 PRINT "15)   ooo  30)  444
45)  JJJ  "
 110 PRINT AT VAL "20",VAL "0"; I
NK VAL "7"; PAPER VAL "2";"------
------------------------"
 115 PRINT AT VAL "21",VAL "1";"E
NTER # OF WAFER INDEX TO LIST"
 120 INPUT A$
 125 IF A$="" THEN  GO TO 25
 130 IF VAL A$<VAL "1" OR VAL A$>
VAL "45" THEN  GO TO VAL "25"
 140 LET A$="@"+A$
 145 LOAD A$
 150 STOP
9900 SAVE "@1,MENU" LINE 20: VERI
FY "@MENU": GO TO VAL "25"
9910 REM
9920 RANDOMIZE USR 100: SAVE "waf
drv.B1" LINE 1
```

```
  10 REM Wafer Index Ver 2.0
  20 REM 1985 RMG SOFTWARE
  25 POKE VAL "23658",VAL "8": BE
EP VAL ".4",VAL "10": INK VAL "6"
: PAPER VAL "1": BORDER VAL "1"
  30 CLS : PRINT  INVERSE VAL "1"
;"******************************
*"
  40 PRINT  INVERSE VAL "1";"****
"; INVERSE VAL "0";TAB VAL "28";
INVERSE VAL "1";"****"
  50 PRINT  INVERSE VAL "1";"****
*******************************"
  55 REM *INSERT WAFER NUMBER *
       ENTER NUMBERS IN LINE 58
    FILL IN BLANKS WITH TITLES
      OF WAFER IF DESIRED.
  58 LET N$="2,1"
  60 PRINT AT VAL "1",VAL "6";"WA
FER #  1 DIRECTORY"
  65 PRINT AT VAL "3",VAL "0";" 1
)              16)              "
  70 PRINT " 2)                17)
          "
  75 PRINT " 3)                18)
          "
  80 PRINT " 4)                19)
          "
  85 PRINT " 5)                20)
          "
  90 PRINT " 6)                21)
          "
  95 PRINT " 7)                22)
          "
 100 PRINT " 8)                23)
          "
 105 PRINT " 9)                24)
          "
 110 PRINT "10)                25)
          "
 115 PRINT "11)                26)
          "
 120 PRINT "12)                27)
          "
 125 PRINT "13)                28)
          "
 130 PRINT "14)                29)
          "
 135 PRINT "15)                30)
          "
 140 PRINT  INK VAL "6"; PAPER VA
L "2";"\\\\\\\\\\\\\\**/////////
//////"
 145 PRINT "*ENTER ""A/C"" - EDIT
 *""S"" - SAVE*"
 150 PRINT "  * FOR ANOTHER, ENTE
R ""FWD"" *"
 155 INPUT A$
 160 IF A$="A/C" THEN  LIST 55: S
TOP
 165 IF A$="FWD" THEN  LOAD "@MEN
U"
 170 IF A$="S" THEN  SAVE "@"+N$
LINE 1: VERIFY "@"+N$(3 TO ): GO
TO VAL "25"
 180 IF A$<>"A/C" OR A$<>"FWD" OR
 A$<>"S" THEN  CLS : GO TO VAL "2
5"
 190 STOP
 200 REM
 210 RANDOMIZE USR 100: SAVE "waf
ndx.B1"
```

LARKEN LINES

by Rod Gowen

By this time I would hope that most of you Larken users would have mastered the basics of your system, but just in case, I am going to take a quick run-through on the basic commands.

FIRST! And most important--if you wish to use the PRINT #4: command, you must enter the following direct command or install it in your BASIC program:

RANDOMIZE USR 100: OPEN #4, "dd"

That command opens channel #4 to the disk drives. Once that line is executed, you need only use the PRINT #4: in front of each of the commands in LKDOS. Those commands are:

PRINT #4:

```
LOAD "fn.B1"                for BASIC
LOAD "fn.C1"CODE          for MC
LOAD "fn.C1"SCREEN$       for SCREEN$
SAVE "fn.B1"              for BASIC
SAVE "fn.C1"CODE          for MC
SAVE "fn.C1"SCREEN$       for SCREEN$
```

You can also SAVE and LOAD all types of ARRAYS, the same as you would with cassette.

```
MERGE "fn.B1"  for BASIC
CAT "",        to CATALOG a disk
CAT ".B1",     Catalogs only BASIC
CAT ".C1",     Catalogs only MC
ERASE "fn.B1", to ERASE Basic
ERASE "fn.C1", to ERASE Basic
```

Beyond these, with the LKDOS Cartridge, you get LKDOS EXTENDED BASIC commands. We will go into these in a later column.

A sample routine would perhaps be helpful. The following sample shows how to set up a Basic program to accept the PRINT #4: and to LOAD and SAVE files:

```
 100 REM SAMPLE PROGRAM
 110 PAPER 1:INK 7: BORDER 1:CLS
 120 RANDOMIZE USR 100: OPEN #4,
"dd"
 130 PRINT #4: LOAD "SAMPLE.C1"S
CREEN$:PRINT #4:LOAD "SAMPLE.C1"
CODE 50000:PRINT #4:LOAD "SAMPLE
.B1"
9999 PRINT #4:SAVE "SAMPL1.B1" L
INE 100
```

In this short sample program, LINE 120 opens the channel to the disk drive, LINE 130 shows the various LOAD commands in actual use. LINE 9999 is a standard SAVE line set up for LKDOS. PLEASE NOTE: YOU ARE LIMITED TO 6 CHARACTERS IN THE FILE NAME + A 3 CHARACTER EXTENSION. File names with more than 6 characters will return the error message "INVALID FILE NAME".

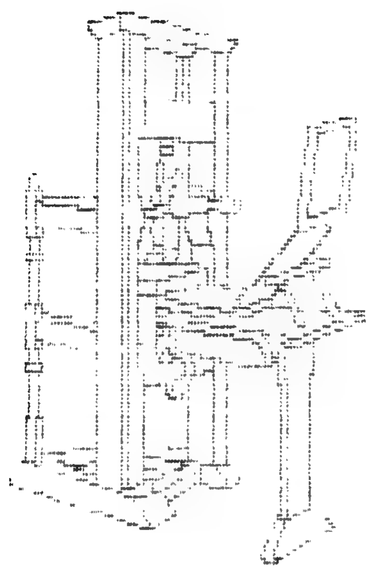
The MERGE command in LKDOS is not quite the same as it is in regular BASIC. It does the same thing, except, that the program continues to run. It does not, as with a normal 2068, MERGE the program and then stop.

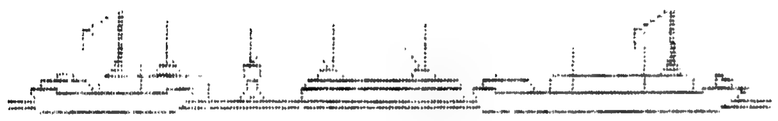
The FORMAT and COPY commands on the Larken system are LOADed from disk. There was just not enough room in the EPROM for them. They are very good routines and work very well.

Larry Kenny says that he will soon have the disk editor finished and ready for sale. We will keep you informed.

Later...Rod

```
   1 LIST 1000: STOP
  10 REM 42 TRACK CONVERTER FOR
    THE AMDISK III & LKDOS BY
   RICHARD HURD
  15 GO SUB 290
  20 POKE 23658,8: REM CAPS ON
  30 CLS : PRINT "1) CAT"'"2) CON
VERT"''"CHOOSE ONE"
  40 IF INKEY$<>"" THEN  GO TO 40

  50 IF INKEY$="" THEN  GO TO 40
  60 LET A=CODE INKEY$: IF A<49 O
R A>50 THEN  GO TO 30
  70 CLS : RANDOMIZE USR 100: CAT
 "",
  80 IF A=49 THEN  PRINT #1;AT 0,
5;"PRESS SPACEBAR FOR MENU": PAUS
E 0: RUN
  90 LET save=40015: REM To save
    buffer to disk
 100 LET load=40012: REM To load
    disk into buffer
 110 LET sprt=40009: REM send
    drive data to buffer
 120 LET setrk=40003: REM move
    head
 130 LET trac=43300: REM trac #
    MC variable
 140 LET drv=43301: REM drive #
    MC variable
 150 REM LOAD TRACK
 160 POKE trac,0: POKE drv,2: RAN
DOMIZE USR setrk
 170 POKE drv,2: RANDOMIZE USR sp
rt: PAUSE 20: RANDOMIZE USR load
```





```
 180 REM Alter Directory Data
 190 BEEP .1,10
 200 POKE 45021,42: POKE 45104,80
: POKE 45106,82
 210 REM Save Track
 220 POKE trac,0: POKE drv,2: RAN
DOMIZE USR setrk
 230 POKE drv,2: RANDOMIZE USR sp
rt: PAUSE 40: RANDOMIZE USR save
 240 CLS : RANDOMIZE USR 100: CAT
 "",
 250 PRINT #1;AT 0,0;"Another? Ye
s or No."
 260 IF INKEY$="" THEN  GO TO 260

 270 IF INKEY$="Y" THEN  RUN
 280 STOP
 290 BORDER 5: INK 0: PAPER 6: CL
S
 300 PRINT TAB 12;"CONVERTER"'''"4
0 Track to 42 Track Conversion fo
r the AMDEK AMDISK III FDD."
 310 PRINT '"This routine LOADs t
he DirectoryTrack into RAM, conve
rts it to  42 tracks and SAVEs it
 back to  disk."
 320 PRINT '"Work is done on Driv
e #0 ONLY!"'''"Make sure that the
Write ProtectTab is OFF!"
 330 PRINT '"You Must First Have
The CODE in the computer to opera
te this    program. This will bre
ak and letyou do that. When you h
ave it in";
 340 PRINT "do this:GOTO 350 to r
eturn to   the GOSUB that this no
te is     printed from. That shou
ld take  you back into the progra
m. If itdoesn't work, you will ha
ve to  GOTO 20";
 345 STOP
 350 PRINT ''''TAB 3;"PRESS SPACE
BAR TO CONVERT"
 355 REM This is the logical
     spot to LOAD the CODE and
   you may want to alter the
  program to do so.
 360 PAUSE 0
 370 RETURN
 380 STOP
 400 RANDOMIZE USR 100: LOAD "Fco
de.CL"CODE : CLEAR 39999: RUN
 410 STOP
 500 RANDOMIZE USR 100: SAVE "40-
42.B1": RANDOMIZE USR 100: SAVE "
Fcode.CL"CODE 40000,1000
 550 STOP
 950
1000 REM Editor's Note-I think
     the CODE needed for this is
   the one I have on disk
  named "Hcode.CL" CODE 40000
  ,2000
1005 REM This program is not
     tested for 2 reasons-it is
   for the AMDEK AMDISK III
  FDD and I think it needs
 the full 2000 bytes as my
own FORMAT program has.
1020
1030 REM The article in Vol 5,
     #10 doesn't say which CODE
   and and the SAVE line only
  indicates 1000 bytes. This
 may be an error. The one I
refer to is in the FORMAT       p
rogram I have on RAMDISK       wh
ich has the ability to        FOR
MAT in 42 tracks also.
```

HARDWARE
A LONG CABLE FOR THE 2040 PRINTER

The short cable between interface and printer has been a major complaint with the Timex printer. I decided something had to be done to fit the printer into my limited desk space. Now I have a 3 ft. replacement cable to permit better printer placement. I elected to make a cable replacement because it was easier than making an extension cable between computer and interface. The display of this modification at the April User Group meeting drew many questions of how to do it. This article is an explanation of the steps I used.

Essential tools: Phillips screw driver, 27-30 watt soldering iron (no soldering gun!); small diameter resin core solder, wire cutter, solder sucker, needle nose pliers, knife, wire insulation stripper. Most of these can be purchased at Radio Shack. Also, a heavy dull pint needle, small vise and two , x 3/4 bolts and nuts.

Material: 7 wire shielded cable, color coded multi-strand wire, I use cable with this identification: E83208 AWM STYLE 2464. The diameter is important, not over .225 inch.

Operation: remove 2 screws in the interface base. Note: where the machine screw was removed. Disassemble interface and make a sketch of the 8 holes where the cable is soldered. Orient the sketch by showing the IC. Number the holes and tabulate wire colors. Unsolder the cable. Remove excess solder at each hole with solder sucker and be sure holes are clear and there are no solder bridges between pads. Use the needle to help smooth the solder at hole edges.

Remove the 2 ceramic radio interference tubes from the cable. Do not separate from sponge pad. Measure the length of wires beyond the end of insulation sheath and the length from strain relief bushing to end of sheath. Make a sketch of these dimensions. I use .60mm to strain relief and 20mm for wire length.

The printer is next. Remove 4 screws from the base and carefully turn right side up. Separate top from base. Carefully lift strain relief and cable from slot. Lift printer circuit board from base (don't lift by printer mechanism). Bolt the mechanism with two , x 3/4 bolts and nuts at opposite corners. Don't be fooled by the mechanism not appearing to be loose; they will separate before you finish and could cause damage.

Unsolder the wires and clean off excess solder at each hole. Be sure holes are clear. Try a needle at each hole, from the solder side. Twisting the needle may help clear rough edges.

Make a sketch of the row with 9 holes where cable is located, plus the ground wire hole. Orient your sketch and number the holes. Make a table of numbers and corresponding wire colors

Make a sketch of cable relief location and length of wires. I use 48mm for wire length and strain relief is at the end of insulating sheath.

To remove strain reliefs, do this carefully...use needle nose pliers to compress sides of sleeve at various locations. The slotted thin end should slide on the sheath when pushed. Next, work the solid end by pushing each way with pliers; using the groove and end. If necessary, slide the needle point between the sheath and strain relief, using great care because the groove has thin walls. Force the strain relief off with plier tips in the groove.

Cut cable to length. You will lose about 5 inches for cable inside equipment. I use 36 to 40 inches of cable. Carefully make cuts around the sheath to expose wire at required dimensions. Do this lightly as the shield is foil with a 7-wire conductor. Bend the cable each time to expose the shield if the cut is deep enough. The sheath will slide off. Trim shield away, being careful of the conductor.

Remove about 1/8 inch of insulation from each wire, don't nick any wires as they can break. Smooth each end so wires are in place and nice and tight. Put a touch of solder on each, only enough to hold wires together. Too much will make wires too big to go in circuit board hole. Do the same with the ground wire.

Now compare the cable wire colors with the original charts. If they are not the same, not the color substitutions. The cable I used has white in place of yellow.

Now install strain reliefs. Note the groove has a flat. This will go toward the case half without a cut-out. The interface end has the flat down while the printer end has the flat up. It isn't mandatory, but the cable lays better this way. Moisten the sheath and carefully force the strain reliefs into position according to you sketches.

Solder cable wires in correct holes, being very sure none of the little wires spread out and don't go in the holes. They can short adjacent pads. Check with a magnifying glass, if any look questionable. Don't forget to install the 2 ceramic sleeves on the cable that fits the interface before soldering.

To assemble the printer, remove the 2 bolts and place the circuit board on the base, fitting the 4 plastic supports properly with the rubber bushings. Press cable and strain relief in the cable slot, place cover over the base, turn over and install 4 screws. Start screws with fingers and then screw driver.

To assemble interface, be sure the 2 ceramic sleeves drop in the slot along with the cable and strain relief. Place bottom cover over the parts, being sure the circuit board and grounding spring is properly located. Install machine screw in the end that holds the spring, and the coarse thread screw in the opposite corner. Now test for proper operation.

WAFER TIPS

by Rod Gowen

This month I will attempt to explain in the simplest way I can, to show all you new microdrive owners how to convert your VU-FILE program to the wafer.

FIRST, AND MOST IMPORTANT, I ADVISE ALL OF YOU TO GET A COPY OF THE HEADER READER PROGRAM FOR THE 2068! This will soon be available to all in the library for you to copy. Our thanks to Dennis Jurries for putting together this utility tape for us.

With the HEADER READER loaded into the 2068, you are ready to convert almost any program to the wafer. The reason you cannot convert VU-FILE without the READER is that it was SAVEd with a direct command and without a loader program separate from the main program. We need to know the number of bytes of MC that we have to SAVE. In this case, we will get readings on the SCREEN and the MC as well as the basic portion of the program.

```
 50 BORDER 1: PAPER 1: INK 1: C
LEAR 28287: LOAD ""SCREEN$: LOAD
```

""CODE

The above listing is what you will see when you start LOADing VU-FILE and hit the BREAK key after the first section after the title appears has LOADed. Now we must change LINE 50 to look like the listing below.

```
  50 BORDER 1: PAPER 1: INK 1: C
LEAR 28287: LOAD "@vf": CODE
```

If you do not care to have the SCREEN$, then you can DELETE that part of the LOAD line. I have found that, with the MICRODRIVE, the SCREEN$ come and go so fast that I don't need them. They do use memory. For the rest of this article I will disregard the SCREEN$. If you want to leave it in, let me know and I will tell you how.

Once you have made the changes, SAVE the LOADer by using the direct command: (make sure you have a formatted new wafer in the drive.)

```
SAVE "@1,vf"LINE 50
```

At this point, you must CLEAR the computer by turning it off and on again. Then LOAD the HEADER READER program. With the program running, start LOADing the VU-FILE tape and watch as it reads the headers. When it gets to the main program code, you will see the following information:

```
................................
....  BYTES:  c DATA LENGTH:  7216
START ADDRESS: 28288
```

This listing gives you all of the information that you need to SAVE the CODE on the wafer. The first line tells you the name of the program, in this case, "c". As this is the machine code portion of the program, it is listed as "bytes". The second line tells you the number of bytes in the program. This one has 7216 bytes. The third line has the starting address of the program. To SAVE this section of the program, you must use the direct command as follows:

```
SAVE "@2,c"CODE 28288,7216
```

Once this is done, you should have a wafer version of VU-FILE. You could try to VERIFY the parts of the program, but I have found that, with the speed of the MICRODRIVE, it is easier to turn off the computer, turn it back on, and LOAD the program. If all was done properly, the finished product should work fine. If you find that I need to be corrected, please do not hesitate to let me know.

CORRECTING GRAPHICS ON PRINTERS

by Dick Wagner

Those members who have recently purchased a printer may be dismayed to learn that circles may print egg shaped on the 2040. There is no correction for the 2068 command CIRCLE, but if you use a formula and you solve the problem. Use this formula to generate a controlled circle on the 2068.

```
  10 FOR I=0 TO 2* PI/200
  20 PLOT 125+80*COS I, 80+80*SI
N I
  30 NEXT I
```

On my 2040 printer the image is 63mm high x 51mm wide. This makes a ratio of 1.24 so COS needs to be 1.24 times wider. Just change to 1.24*80* COS i to print a practically true circle. You could also use 80/1.24*SIN and have a 51mm circle.

A similar problem may exist with the TV screen and can be corrected the same way. My TV has an image 15% higher than wide.

10 Years Later

A Retrospective

10 Years Later- -A Retrospective

by: Rod Gowen

Has it really been 10 YEARS!! I guess the old adage is true--"time flies when you're having fun". It really does not seem possible that it was 10 years ago, Sept. 1991, when Bob Evans, Dennis Jurries and a couple of others met officially for the first time. Our first meeting place was on the corner of Molalla Avenue and Beavercreek Road in a former fireplace shop that was being used at the time by one of our first members, Quentin Rippey, for his foam rubber and styrofoam business. How many of you current members can remember this? How many of you remember Dennis Jurries, our co-founder? Do you know that we still have 3 of the first 4 members actively participating in our club? If you didn't, they are Bob Evans, membership #2, Rod Gowen, #3, and Dick Wagner, #4. I wonder how many other TS user groups can say the same? I wonder how many of the existing clubs are 10 years old?

The Beginnings. .

But, I am getting a bit off the subject. In this case, a look back on my 10 years with this great group! I can honestly say that I have enjoyed every bit of the time spent with our members. I can, as acting Treasurer of the group, say that I have met every one of the 87 people that have passed through our little group on their way through the ever-changing world of computers. I have collected dues from almost all of them. I will say that our first secretary/treasurer, Don Paul, did a fine job for the first year or so of our existence. Don, by the way, was, to the best of my knowledge, our only British born member. Our members' ages have ranged from 12 years of age to 76 years young. I think that the average age of our members over the years must be around 50-55 years old. I believe our current paid-up member list numbers about 15-18 die- hard TS supporters.

The Places. . .

A brief history of our group will follow us through our various meeting places and some of our events and high points. As I said earlier, our first meeting place was in an empty fireplace store where we had to bring our own chairs and tables. From there we moved to Clackamas Community College where we met on the 2nd Thursday of the month as the college campus was closed at 5 PM on Fridays during the summer. From there we moved to our present location, the Far West Federal Bank community room in the Oregon City Shopping Center where we have been meeting on the 2nd Friday of the month (except for rare occasions) ever since. Some of the events in our short history lesson have been the dinner meetings that have been held 3 times and the swap meets that have also been held 3 times. We were also fortunate enough to have participated in 3 "mini-fairs" over the years. The first, at the home of the now defunct magazine, Time Designs, in 1986. The second, in Seattle, in 1987, and the final one (and the biggest) in Portland in 1988. This last, THE 3RD ANNUAL GREAT NW TS MINI FAIR was co-produced by RMG Enterprises and Time Designs Magazine and was hosted by our group. A lot of group effort and cooperation went into some of these events and I feel that they, along with the continued comradery, have made being a member of CCATS well worth the effort.

RMG. . .

As most of you know, I own and operate RMG Enterprises, perhaps the world's largest remaining TS dealer. It may interest you to note that the formation of the computer portion of RMG runs parallel to that of CCATS. I ordered my first Sinclair ZX-81 from England in the fall of 1980 and received it just after Christmas of 1980. I then started looking at sources for software and impatiently awaited the release of the forthcoming printer and other promised add-ons. When, after 6 months of experimenting and learning, I learned of a local man doing the same thing I was doing, and, as the Timex Corporation had announced plans to market the Timex Sinclair 1000 in the U.S., was actively promoting the new $99 computer in local stores, I contacted him. Dennis Jurries was his name. He had already been contacted by Bob Evans and this was the basis for our present group. I started out by buying up items that the members of the group were interested in and could assure them of a lower price if i were to buy in bulk. I worked a full-time job and did the computer business on the side until 1985 when I was laid off from my 15 year job and decided to take a chance and go with the mail-order computer business full- time. It worked out well for both CCATS and RMG. In todays market I could never make a living selling exclusively to the TS market. RMG has broadened its base to include IBM clones and related peripherals in order to keep ahead of the bills. The group, as individuals and as a whole has had a slight edge due to having RMG as a member and RMG had a small but dedicated built-in customer base. I have appreciated the business over the past 10 years and I am sure (folks have told me so) that the members of the group have enjoyed having a "local dealer" instead of having to do everything by mail.

The Newsletter. . .

Our fine newsletter, THE PLOTTER, is not quite as old as the group. It premiered in September of 1982 and has been in continuous publication ever since. It has been published 11 times per year minimum for 9 years. The first issue was a 2 page affair with no name or graphics. It was printed on my first Gorilla Banana 80 column printer using Memotext on my ZX-81 with the Memotech Centronics interface. We had a name by the second issue and were off and running. It slowly grew and evolved over the years. It stabilized at 8 pages and I edited and published it with some help from Dennis Jurries until 1985 when our current editor, Dick Wagner, took over. I, for one, applaud him and congratulate him on a job well done for the past 6 years! The job of newsletter editor, as any of them can tell you, is a thankless job at its best. With reader input always low, it can take a lot of time to do a credible job of putting together a publication that is both informative and entertaining.

The Computers. . .

The last part of my "group history" has to be the history of the computers that we have used and loved for the past 10 years themselves. As I noted above, some of us started out with the second computer that Sir Clive Sinclair put on the market, the ZX-81. I venture to say, however, that the majority of our members have started with the TS 1000. Sir Clive started with the ZX-80 in the fall of 1979 and went to the ZX-81 in the fall of 1980. He followed that with the ZX-Spectrum and ultimately, with the Sinclair QL (Quantum Leap). His historic agreement with Timex to manufacture and market the U. S. version of the ZX-81, to be known as the TS 1000 gave us the push we needed to get our group going and to sustain it. Timex followed the 1000 with the TS 1500 which was basically a TS 1000 in a silver Spectrum case, has the advantage of having 16K of ram onboard and a much better rubber key keyboard. This unit made its appearance in late 1982 to be followed within 2 months by the much heralded TS 2068. This was to be the "ultimate" $200 computer! With 72K of ram (37.5K usable), sound, color and high resolution graphics it sure looked like it! The 2068 finally arrived in stores and at RMG in late November of 1982 and was selling quite sell. Little did we know that 3 months later Timex would perform the ultimate "dump" on all of us who were so avidly supporting their computers! The first week in February of 1983 Timex Computer Corp. announced that they were calling it quits and that was that!

Things were far from over though. There were many loyal dealers and third party software/hardware producers who were to stand by us for several more years. Some saw the Timex pull-out as the end and quit within weeks of Timex. The ones that remained were the basis of a long and, for most of us, enjoyable life with our computers. Almost EVERY major peripheral for the 2068 appeared AFTER Timex had already quit!

A company in Texas called AERCO (Acme Electric Robot Co.) was the first to produce and market a disk drive system for the ZX-81/1000 and the 2068. Another company called COMPUSA also had a disk drive for the ZX-81/1000. Another company in California, A&J MICRODRIVE, came out with small, less expensive "stringy- floppy" high-speed tape drives for the ZX-81/1000 and the 2068 about the same time as Aerco's disk drives. This was based on the older CAI String Floppy system from England. Before long, we had more disk systems available from LARKEN ELECTRONICS, THE JOHN OLIGER CO. and even an import from Portugal in the form of THE TIMEX TOS DISK SYSTEM. This was to have been the disk system that Timex would have marketed in the U.S. for our computers. The only major difference in the Portuguese version was the fact that their 2068 had a Spectrum compatible bus. In the years since, the clear-cut system of choice seems to be the LARKEN LKDOS system. LARKEN also developed a 256k ramdisk and had the foresight to produce code on a DOS cartridge to enable most of the other disk systems to use his DOS (Disk Operating System).

The first modem available for the TS computers was produced by BYTE-BACK CO., the MD-2B for the TS1000, to be followed by the first one for the 2068 called the MD-68. Even though Timex was ready to release the TS-2050 modem to work with all of the TS computers, at the last minute they refused to take delivery of the 100,000 units that a company called Westridge was waiting to ship when they called it quits. It was almost 6 months after the demise of Timex Computer Corp. before we saw the promised Timex 2050 modem hit the market under the name of the producer, WESTRIDGE. As a 300 baud modem with no software to allow for downloads on the 1000, this was a limited but effective modem. It wasn't too long before folks were looking for an alternative. Enter the need for an RS232 port.

Byte-Back was the first with an RS232 for the 1000 and then for the 2068. Problem was, they were one of the first to quit within a year or so of Timex quitting. Aerco had and still has a dual RS232 port available for the 2068 but it had a very high price tag: $100! A schematic appeared in Time Designs Magazine that told how to convert one of the thousands of 2050 boards to an RS232 port. A company called ED GREY ENTERPRISES in Los Angeles put out the ZSI/O Port as a kit and it has long since sold out. At the time of this writing the only commercially available RS232 port is the AERCO dual port. There have been rumors of a new one coming from Larken, but we will not hold our breath. With the use of these ports we were able to make use of the 1200 baud modems and make telecommunications more efficient. There are still several BBS systems around the country for the TS users, as well as a forum on Compuserve.

Aerco produced one of the first parallel printer interfaces for the ZX-81/1000 and the 2068. A British company, MEMOTECH, set up a base of operations in Denver, CO, and sold a wide variety of hardware for the ZX-81 /1000 in this country. A young Canadian, Peter Hacksel, also put out a parallel port in two models: one for the rear bus and one to fit into the cartridge port. It was fully compatible with the Aerco model. In point of fact, the Aerco has been almost universally accepted as the standard for the 2068. There were others: the TASMAN units from England and the OLIGER interface. The OLIGER unit is still available as is the Aerco CP-68. Many printer driver packages have been written and many are still available.

Now that we have 80 column printers, 1200 baud modems and disk drive systems, what more could we want? Well, it seems as though we are never satisfied. There is always something that folks will want. We have had video digitizers introduced as well as voice recognition and voice synthesizer systems. We have had some fabulous software written for the 2068 and well as a lot of the British Spectrum software imported and some of it converted for the stock 2068. As long as we keep active and curious, there will be a few more good years for the 2068. As for the 1000, it is still in strong demand by experimenters and ham radio operators because it is so easy to set up as a control device, a "programmable" controller.

CONCLUSIONS. . .

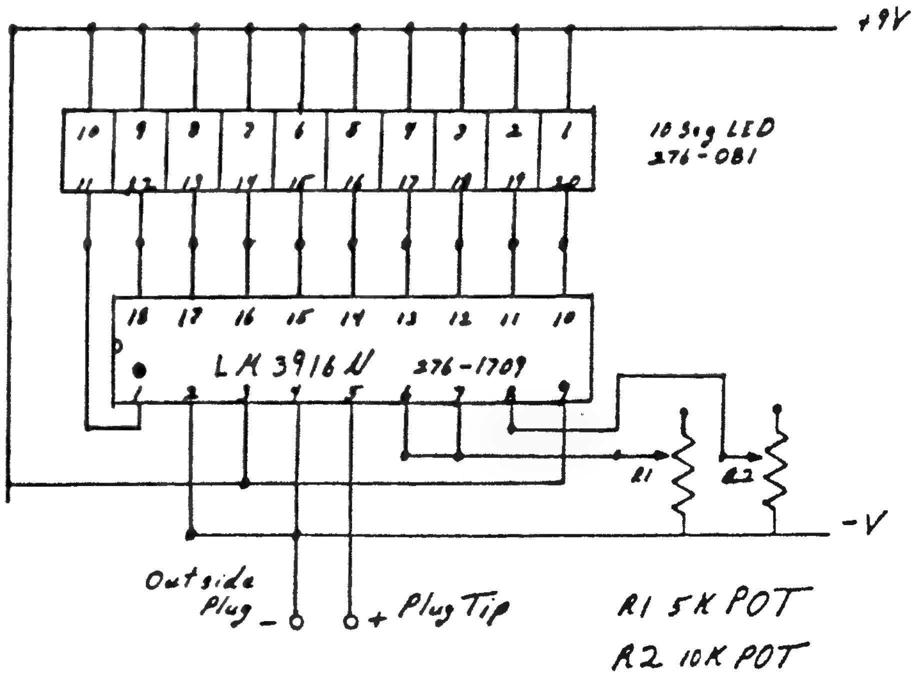
I know that this piece cannot begin to cover the complete story. I have not even begun to list all of the software and hardware that was and is available for the TS line of computers. I have not mentioned the many computer shows and fests that have been held over the years or the many user groups that were and still are active across North America. I do not have the room to talk about all of what has passed before me in the past 10 years. All I am trying to do here is to jog your memories, to try to capture a little of the excitement that has held us together for the past 10 years and will continue to do so in the future. I know that I have enjoyed each and every one of those years and I know that those of you who have been with us for a long while must also have enjoyed yourselves else why would you have stuck around? I am looking ahead and would like to see where we are 10 years from now! I do know one thing for sure: as long as there is a customer out there who wants to buy a TS product, RMG and I will be here to try to help. If there is a CCATS group, I will be a part of it.

WILL YOU?

****

LOAD/SAVE METER

(Continued from Page 97)



Index

QL INDEX:

GENERAL INDEX:

WE'RE NOT PERFECT!

Of course, we never claimed to be. After we had our first 25 copies of THE BEST OF THE PLOTTER printed, we found some errors. Many are simple typos, and there will definitely be more found. We are including this errata sheet to help you with the known errors. If you find any typos or errors, please drop us a card or note so that we can make the changes for the next printing. We will also let the other readers know what has been found. All known errors will also be published in THE PLOTTER as they become available.

PAGE 22: MORE TASWORD II, Line 4;
Change "familiat" to "familiar"

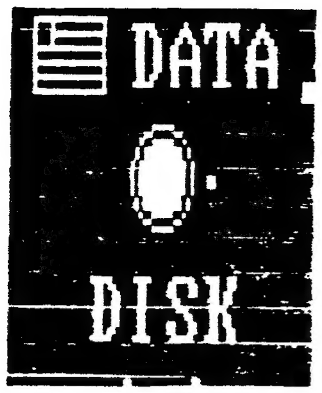
PAGE 26: BUZZ SAW, Line 1;
"graphis" should be "graphics"
Line 3;
Delete one "work"
Line 15;
Change "teeht" to "teeth"
Line 28;(last line of text)
Line should end with "on page 74."

PAGE 31: QUICK UDG PROGRAM, Line 6;
Change "bay" to "way"

PAGE 44: CIRCLES, Line 10;
Change "ypward" to "upward"

PAGE 45: Program line 1010;
"touse" is really 2 words, "to use"

PAGE 76: BAR CODE?, Line 2;
Change "sence" to "sense"



ORDER FORM

To get a disk that contains most of the programs listed in this book, just fill out this order form and mail it, with your check or M.O. for $9.95 to the publisher, RMG ENTERPRISES. Please tell us disk size/sides/tracks. LKDOS or OLIGER formats available.

(PLEASE PRINT OR TYPE)

NAME:______________________

ADDRESS:__________________

CITY/ST/ZIP:_______________

DISK SIZE/SD/TR:___________

PRICE: $9.95 POSTAGE PAID

Gift Copies Of The Best Of The PLotter

If you would like to order extra copies of this book for yourself or for TS friends, just fill in this order form or call RMG at the address in the front of the book.
The price is just $14.95 postpaid.

(Please print or type)

NAME:______________________________

ADDRESS:___________________________

CITY/ST/ZIP:_______________________

PRICE: $14.95 PER COPY POSTPAID